DTIC-BR-TR--2001/01

AD-A385219

The DTIC® Review

Cybernetics: Enhancing Human Performance

Unclassified/Unlimited

Vol. 5, No. 3
March 2001

The* DTIC® *Review is published by the
Defense Technical Information Center (DTIC),
DTIC-BR, 8725 John J. Kingman Road, Suite 0944
Ft. Belvoir, VA 22060-6218
Telephone: (703)767-8266, DSN 427-8266
FAX: (703) 767-9070, DSN 427-9070
Email: bibs@dtic.mil

Opinions expressed herein are those of the authors and do not necessarily reflect those of the Department of Defense or its elements.

Distribution
The DTIC Review is approved for public release.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
03-2001	Final	March 2001

4. TITLE AND SUBTITLE

The DTIC Review
Cybernetics: Enhancing Human Performance
Vol. 5 No. 3

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Cupp, Christian M.; Editor
Levine, Phyllis; Research Advisor
Lahaie, Eric Z.; Compiler

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Defense Technical Information Center
DTIC-BR
8725 John J. Kingman Rd, Suite 0944
Ft. Belvoir, VA 22060-6218

8. PERFORMING ORGANIZATION REPORT NUMBER

DTIC-BR-TR--2001/01

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Defense Technical Information Center
DTIC-BR
8725 John J. Kingman Rd, Suite 0944
Ft. Belvoir, VA 22060-6218

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

A – Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

This publication is published irregularly by the Defense Technical Information Center.

14. ABSTRACT

This edition of *The DTIC Review* explores advancements in the field of cybernetics and the effects this research has on human performance. Cybernetics attempts to blend humankind's ability to think, reason and learn with machine-kind's productivity and efficiency. Cybernetic technology is already used industrially, militarily and scientifically, and its impact will only become more profound in the future. At present, cybernetic technologies are enhancing human performance via training, research assistance, and manual labor.

The DTIC Review is compiled from recent technical reports submitted to DTIC's database. The bibliography is only a representation of available materials.

15. SUBJECT TERMS

Cybernetics, Robotics, Haptics, Artificial Intelligence, Virtual Reality, Human Performance, Virtual Environment, Human Factors Engineering

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON / 19b. TELEPHONE NUMBER (include area code)
Unclassified	Unclassified	Unclassified	Unclassified Unlimited	244	Phyllis Levine / 703-767-8266

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

20010109158

THIS PAGE INTENTIONALLY LEFT BLANK

The DTIC® Review

Cybernetics: Enhancing Human Performance

AD-A385219

Vol. 5, No. 3
March, 2001

THIS PAGE INTENTIONALLY LEFT BLANK

FOREWARD

Cybernetics attempts to blend humankind's ability to think, reason and learn with machine-kind's productivity and efficiency. Cybernetic technology is already used industrially, militarily and scientifically, and its impact will only become more profound in the future. Technological advancements in cybernetics will dramatically affect our lives as the 21st century unfolds.

This edition of *The DTIC Review* focuses upon the latest developments in the area of cybernetics and its benefits for human performance.

The editorial staff hopes you find this effort of value and appreciate your comments.

Kurt N. Molholm
Administrator

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

INTRODUCTION

Cybernetics attempts to blend humankind's ability to think, reason and learn with machine-kind's productivity and efficiency. While modern computers and robots have proven to be invaluable tools for people, they still remain limited by programmed parameters and dependent upon human interaction. Cybernetic technologies will slowly remove these barriers and allow the development of machines that think and learn on their own by imitating a person's brain. The end result will be the creation of a virtual human.

Cybernetic technology is already used industrially, militarily and scientifically, and its impact will only become more profound in the future. Some current uses for cybernetics include, but are not limited to, civilian and military training, research assistance and manual labor. Human performance research tools along with virtual environment technology are being used to keep up with the changing training requirements of DoD personnel. More emphasis is being placed on practicing a variety of situations, simulation, scenario based training and situated learning. Ultimately, cybernetic research will provide people with higher levels of safety through superior training capabilities and/or a reduction of high-risk tasks that must be performed by humans rather than machines. The addition of cybernetic capabilities to these machines would increase their usefulness exponentially. Technological advancements in cybernetics will dramatically effect our lives as the 21st century unfolds. The blend of human and machine will help develop the skills necessary for the warfighter of the future.

The selected documents and bibliography are a representation of the material available on cybernetics from DTIC's extensive collection. Additional references, including electronic resources, can be found at the end of the volume. In-depth literature searches may be requested by contacting the Reference Team, Network Services Division at the Defense Technical Information Center: (703) 767-8274/DSN 427-8274;
FAX (703) 767-9070; E-mail bibs@dtic.mil

THIS PAGE INTENTIONALLY LEFT BLANK

DOCUMENT 1

Soldier Performance
Course of Action (COA) Visualization Aids

AD-A382305

Aug 2000

Arizona University – Tucson
Department of Electrical and Computer Engineering
Tucson, AZ

THIS PAGE INTENTIONALLY LEFT BLANK

ARMY RESEARCH LABORATORY

Soldier Performance Course of Action (COA) Visualization Aids

Jerzy W. Rozenblit
Michael J. Barnes
Faisal Momen
Jose A. Quijada
Theodore Fichtl

ARL-CR-302 AUGUST 2000

prepared by

University of Arizona
Electrical & Computer Engineering Dept
1230 E. Speedway Blvd.
Tucson, AZ 85721

under contract

DAAD17-99-P-0203

Open Inventor™ is a trademark of Silicon Graphics, Inc.

Army Research Laboratory
Aberdeen Proving Ground, MD 21005-5425

ARL-CR-302 August 2000

Soldier Performance Course of Action (COA) Visualization Aids

prepared by

Jerzy W. Rozenblit
Faisal Momen
Jose A. Quijada
University of Arizona

Michael J. Barnes
Human Research and Engineering Directorate, ARL

Theodore Fichtl
Compass Foundation

under contract

DAAD17-99-P-0203

Abstract

The computer revolution has resulted in extending the possibilities of battlespace visualization to the brigade commander and below. However, mobility and bandwidth considerations require that the systems be efficient to reflect the realities of modern combat. The Advanced Battlespace Architecture for Tactical Information Selection (ABATIS) is being developed to be a rapid planning and re-planning experimental environment. ABATIS's object-oriented architecture has the advantage of being able to rapidly construct a three-dimensional battlespace that will accurately represent the essential planning components of a brigade and smaller division battle environment. The basic architecture has been extended to include war-gaming logic as part of the software design, and examples are given that pertain to specific military problems. This capability will allow ABATIS to realize fully the implications of battlespace visualization by creating a human-computer synergy that encourages both human and machine to generate and evaluate possible courses of action and their consequences. The human performance implications are discussed, and particular attention is directed toward research issues related to terrain visualization, automation, decision making, and cognitive biases.

TABLE OF CONTENTS

INTENTIONALLY LEFT BLANK

SOLDIER PERFORMANCE COURSE OF ACTION (COA) VISUALIZATION AIDS

1. Introduction

The human's ability to visualize complex problem spaces is an important part of both scientific and military lore. Ulysses S. Grant, for example, could not only visualize minute details of the impending battle area but could actually envision troop movements and bottlenecks while planning his tactical maneuvers (McPherson, 1999). The purpose of this research project is to extend this capability via modern computer technology that symbolically abstracts the most important features of the battlespace, including the behavior of U.S. and enemy forces. The research focus is the cognitive and perceptual performance of the combined human-computer system. To support the research program, the authors created a specialized software system called "Advanced Battlefield Architecture for Tactical Information Selection" (ABATIS) (Keane, Rozenblit, & Barnes, 1997). ABATIS is a three-dimensional (3-D) visualization system that facilitates rapid, flexible development of high-level battlespace representations as well as execution and assessment of war-gaming scenarios.

This report discusses recent refinements of the ABATIS system, which will eventually extend the visualization domain into human-computer research paradigms via intelligent algorithmic modules. The refinements follow directly from the meaning of visualization that implies understanding the process and "end states," not simply presenting finely grained detail of the physical world (Barnes, 1997). The extensions of ABATIS will allow the quick creation of new tactical environments, investigation of optimal U.S. and enemy end state behaviors, and better understanding of the human role in this symbiotic environment.

Our focus is narrowed to the interplay of tactical decision making, situational awareness, and the continuous planning process via intelligent aiding. Our concerns are the cognitive problems associated with visualization and operator performance in automated planning environments, particularly, in situations when the planner must address multiple sources of uncertainty. Recent analyses of automated systems indicate that the extent to which human operators mistrust or conversely over-rely on automated systems depends on their state of situational awareness (Parasuraman & Riley, 1997).

The principal issue is the ability of the human to understand enough about the planning scenario and the behavior of the intelligent systems to make well-informed supervisory choices without losing insight into the unfolding battle trends. Intelligent systems can remove the operator from the decision process and inadvertently create a situation in which the human can no longer react to new developments. Conversely, the human may not trust the computer solution and may choose to follow his or her own instincts when they are inappropriate. We hypothesize that both situations have the same root cause (the inability of the decision maker to visualize the broader military context while understanding the implications of the suggested courses of action [COAs] proposed by the automated system). Using ABATIS, we intend to investigate better visualization techniques whose purpose is to impart insight as well as suggested decision options during the planning and re-planning process. Our research goal is a human-computer synergy that decreases planning time while maintaining the intuition and insight of the human component through combining the explanatory power of visualization with the computing power of intelligent systems. We intend to establish the utility of ABATIS as a research tool and as an early prototype of a versatile planning and re-planning tool for "brigade-and-below" applications.

2. Background: Human Performance Issues

2.1 Terrain Visualization

ABATIS supports a 3-D perspective military terrain generator that can be viewed from multiple angles and perspectives. The 3-D effects are produced by renderings that depend on perceptual factors such as volume, perspective, shading, and relative size to produce the desired effects. A variety of issues related to terrain visualization was investigated by the University of Illinois researchers (Banks & Wickens, 1999; Wickens, Thomas, Merlo, & Sehchang, 1999). The two principal foci of this research were the effects of visualization dimensionality and viewpoint. A common assumption among display designers is that 3-D perspectives are the preferred presentation mode for military terrain because these perspectives are similar to the natural world. However, converting 3-D information onto a two-dimensional (2-D) display plane introduces perceptual ambiguity because of foreshortening and resolution losses in the depth dimension. For example, a number of experiments investigating aircraft display formats indicate poor resolution in the altitude dimension for air traffic control tasks whenever the observers were using 3-D as opposed to 2-D representations (Merwin, O'Brian, & Wickens, 1997). Other problems related to altitude and azimuth determinations have been noted for navigational tasks that required 2-D map to 3-D scene translations (Schrieber, Wickens, Goetz, Alton, & Hickox, 1998).

In an extensive survey of aircraft-related research, Banks and Wickens (1999) found many cases in which 2-D display representation was superior to the higher dimensional representations and vice versa. Based on these findings, they investigated military map problems using U.S. Military Academy cadre as subjects to investigate the following map tasks: assessing mobility corridors, relative position judgments, and line-of-sight (LOS) determinations. Again, the relative advantages of dimensionality were highly task dependent; only the LOS tasks showed any clear advantage for the 3-D conditions. The other variable that they investigated was the degree of exocentricity (i.e., the relative distance of the viewer above the scene). Extreme exocentric conditions involved a bird's eye view of the terrain, whereas the closer egocentric conditions involved an immersed view as if the operator were observing the terrain from a low altitude.

In the immersed conditions, the observer could move freely within the terrain boundaries. The results were similar to dimensionality results in that the advantages of viewpoint depended on the particular military task and dependent measure. For example, LOS tasks resulted in more accurate LOS determinations for immersed views but at the expense of increasing the total time spent performing the task. In an ensuing study, Wickens, Thomas, Merlo, and Sehchang (1999) focused on potential cognitive problems associated with being immersed within the terrain scene. Again, using U.S. Military Academy cadre, they discovered a cognitive tunneling effect for the immersed condition. This effect resulted from subjects' inattentiveness to important military events occurring to their rear in the immersed map environment.

Other researchers investigated similar viewing factors via a more abstract scientific data visualization paradigm. When the observer was required to navigate and make relational judgments in 3-D data space (McCormick, Christopher, Banks, & Yeh, 1998), degree of exocentricity was an important factor. However, the results were not monotonic; intermediate views (half way between immersed and bird's eye) actually resulted in slower search performance than either extreme. Apparently, this view had neither the advantage of the proximity of the immersed view nor the overall contextual superiority of the exocentric view. In general, the results followed the expected pattern: tasks that required local judgments were better supported by immersed views and those tasks that depended on global cues were better supported by exocentric views. Wickens, Merwin, and Lin (1994) investigated the effects of dimensionality on information integration tasks. Three-dimensional representations resulted in better integration among the cognitive dimensions of price, debt, and earnings as opposed to 2-D planar representations (requiring integration over two displays) of the same information. Also, stereopsis (ocularly induced as opposed to 3-D renderings) aided in information integration. Interestingly, the 3-D performance gains were not evident during ensuing memory tasks.

There are three ways to produce 3-D effects: perspective renderings, stereopsis (based on binocular effects of retinal disparity), and motion induced (Kaiser & Proffitt, 1992). These 3-D factors act in concert with each cue that contributes to the scene's realism as an additive weighted component (Sollenberger & Milgram, 1993). Stereopsis and motion-induced effects improve performance of certain tasks (Barfield & Rosenberg, 1995; Yeh & Silverstein, 1992), but they have their own set of problems that are beyond the scope of this research effort (Mon-Williams & Wann, 1998; Patterson, Moe, & Hewitt, 1992). Our initial efforts concentrate on 3-D rendering cues and the results will be used to develop overall guidelines for the use of viewpoint (viewing angle and immersion factors) and dimensionality to enhance tactical decision tasks (Barnes & Wickens, 1998). The results will delineate how best to use the versatility of ABATIS to accurately portray military scenarios within a process-centered environment.

2.2 Tactical Decision Making

For the most part, this research studied perceptual and cognitive effects related to situational awareness (Endsley, 1995). ABATIS is being designed to investigate the synergy between computer visualization and artificial intelligence and their combined effects on the war fighter's tactical decision making. Other researchers have concentrated on the soldier performance effects of combining these two components (Marshak, Winkler, Fiebig, Stein, & Khakshour, 1999), and important research continues in visualization factors related to soldier immersion and dimensionality (Wickens, Thomas, Merlo, & Sehchang, 1999). However, more research needs to be done which focuses on the relationship of human uncertainty to automation. A recurring problem with automated systems is trust (Parasuraman & Riley, 1997). In particular, early decision-aiding approaches tended to be sophisticated in a technical sense but naïve in a practical sense; experts did not know when to trust them.

This lack of understanding of the computational processes of intelligent systems can lead to two seemingly unrelated system deficiencies: complacency and mistrust. Both conditions result at least in part from the human viewing the intelligent algorithm as a separate or even a competing entity. The crucial factor underlying both mistrust and complacency is the lack of insight by the human operator as to exactly what it is the machine is doing over some extended period of time. Unfortunately, the problem is complicated further by the behavioral characteristics of humans when they reason while in uncertain environments. In the last 25 years, a seemingly never-ending list of human biases, limitations, and psychological illusions has been documented in the behavioral decision literature (Kahneman & Tversky, 1973; Einhorn & Hogarth, 1981; Hollands & Wickens, 1999). The usefulness of probabilities is a controversial subject. In the popular book "A Civil Action," for example, the evidential propriety of probabilistic information in general was challenged by the defendant's lawyers (Koehler, 1993). Logically, not assigning a number to an uncertain event does not make it deterministic, and yet probabilistic evaluations of possible future COAs are resisted by military leaders for a number of reasons, not the least of which is the difficulty of generating valid probability values. New systems are being developed which will generate probabilities for possible intelligence outcomes

(Jones et al., 1999; Charles River Analytics, 1998), but the results depend on the ability of trained analysts to generate accurate probabilities. Again, the basic issue is trust. The user of intelligence estimates must trust the intelligent algorithm and the probability elicitation process that feeds the algorithm.

2.3 Poor Calibration of Probability Estimates

The overconfidence phenomena have been documented by a number of researchers (Sniezek & Buckley, 1993; Hollands & Wickens, 1999). The basic paradigm is to ask human subjects to answer a general knowledge question and then state their confidence level. The accurate confidence level should correspond to the overall percentage correct on the general knowledge test. In fact, humans tend to be over-confident by 20% to 30% (obtained score – average confidence level). This phenomenon extends to experts of all types, novices and college students; weathermen seem to be the one of the few groups that is well calibrated. Sniezek and Chernyshenko (1998) recently replicated this phenomenon at the U.S. Army Intelligence Center and Fort Huachuca, Arizona, by using senior retired intelligence officers. The impact on intelligence estimates is obvious; senior officers do not like to be wrong, and yet, the numeric confidence levels they assigned to their answers were consistently overly confident. The other side of the coin is that the operator's use of probability estimates displayed on the computer does not always follow prescriptive decision rules. One such deviation from normative behavior is the phenomenon of probability matching: the tendency of humans to match rather than optimize probability sequences. This is related to gambler's fallacy and the tendency of the decision makers to be influenced by previous outcomes for independent events. An example from one of the author's personal experiences is the tendency of subjects to override automatic target recognition (ATR) algorithms when it is inappropriate to do so (their performance was actually less than chance). In this particular case, the operator tended to match the stated accuracy level of the ATR as if he or she felt compelled to override the system a certain percentage of the time even though objectively, the operator performance was quite poor in these circumstances. The overall research results suggest that the human operator is poorly calibrated in both using and generating probabilistic information (Barnes, 1979; Hollands & Wickens, 1999). Sniezek and Chernyshenko (1998) have designed research and training stratagems to alleviate the latter problem; our research interests are focused on visualization techniques to improve the user's ability to understand and use probability estimates generated by the computer.

2.4 Confirmation Bias

Many of the biases discovered in the literature are attributable to human processing limitations (March, 1978). Of particular importance in a military setting is the sequence of when information is processed and its effect on decision making. The USS Vincennes incident is a good example of one manifestation of sequence effects. The initial reading of the screen suggested to the radar operator that the incoming plane was descending with hostile intent.

Later evidence indicated the aircraft was neutral and ascending, but the action officer and the commander were looking for evidence of immanent attack, and thus, the initial decision was amplified rather than contradicted as new information was received (Hollands & Wickens, 1999). Adelman, Bresnick, Black, Marvin, and Sak (1996) found a similar overweighing of initial cues for Patriot air defense officers who were more influenced by the action of the incoming aircraft if the action was done early in the sequence as opposed to the same objective pattern with the cues occurring late in the sequence. This seemed to be another example of the decision maker forming an hypothesis early and favoring cues that supported the hypothesis while discounting equally valid cues contradicting it. The problem is more complex than these examples indicate because there are also cases when the opposite occurs. A number of experiments have demonstrated a recency effect; cues that are later in the sequence have more impact than the earlier information even for similar tasks (Adelman & Bresnick, 1992). Hollands and Wickens (1999) argue that the simplicity of the initial cues and the length of the set of updating cues may explain the difference. In the Vincennes incident, the hostile hypothesis was generated early and events occurred quickly. Perhaps in cases when the initial hypothesis is less firmly held and the intervening information unfolds over a longer time period, recency of information outweighs the initial direction of the data sequence. It should be obvious that both instances are valid strategies for overcoming processing limitations, allowing the observer to concentrate on the most crucial information rather than be overwhelmed by the constant data stream. Both tactics have ecological validity. Forming an early hypothesis and collecting data related to the hypothesis are effective means of handling complex data spaces. In combat, changing the hypothesis often may be worse than "sticking to your guns" once you have reached a conclusion unless the disconfirming evidence is strong.

On the other hand, recency effects may be justified in a volatile environment wherein the initial information is no longer valid. In general, the perceived validity of intelligence degrades as a function of time. The sequence in which combat information is received and the early formation of hypotheses concerning enemy intent are important cognitive factors in explaining the relative effectiveness of different combat planning conditions. It will be important to know in particular whether information collected early in the planning process is assigned too much or too little weight as more recent intelligence is collected. This particular problem is expected to interact with validity estimates of intelligence sources and the general problems associated with probability estimation. Both of these issues will interact with visualization; the more graphic and compelling the battlespace image, the more likely the user will be to assign too much weight to probabilistic cues and to prematurely choose a COA that more recent information may contradict. The challenge is to develop visualization principles and feedback techniques that impart insight into the probabilistic nature of the process, including the possibility of abrupt change. The objective of the ABATIS research environment is to understand the effects of these psychological factors in a rapid planning and re-planning tasking for a versatile, highly mobile force.

The following describes the general architecture of ABATIS, future extendibility, and the military context it is being developed to investigate. The overall purpose of the research project is to determine general design principles for these situations, which are based on realistic soldier performance and cognitive parameters.

3. ABATIS

The U.S. military extensively employs simulation-based, virtual training systems known as computer-generated force (CGF) systems (Hancock, 1994; Karr, Reece, & Franceschini, 1997). Such systems incorporate live, virtual, and constructive simulation in high resolution, synthetic environments. The disadvantages of these systems are the complexity of communication protocols they require when used in a distributed setting and high communication bandwidth constraint. By design, they do focus on battlespace abstractions; their goal is to replicate a battle environment in a computer-based system so that training costs can be reduced.

Examples of systems that share some similarities with our visualization environment are JANUS(A)[1] and, more recently, commander's intelligent battlefield information display (CIBID) and virtual geographic information system (VGIS). JANUS(A) is used by the U.S. Army as an interactive, computer-based, war-gaming simulation of combat operations conducted at the brigade and lower levels. It consists of two opposing forces that are controlled by two interacting players. JANUS(A) concentrates on ground combat. It is composed of Army-developed algorithms and data to model combat processes. The program comprises approximately 200,000 lines of legacy code (VAX [virtual address extension]-11 FORTRAN [Formula Translator], a structured Digital Equipment Corporation [DEC] extension of American National Standards Institute [ANSI] standard FORTRAN-77). This aging technology seriously impedes any efforts to implement the concepts required by the commander's post of the future.

The CIBID software architecture currently being developed by CHI (computer-human interaction) Systems, Inc. (Graves & Miller, 1998), is a 2-D battlefield visualization tool that uses object-oriented design principles. Users can work with digitized maps to create a battle scenario via the existing 2-D Army symbology. facilities are provided to execute war-gaming scenarios in a model-based environment.

[1]not an acronym

VGIS allows interaction and navigation in very large, high resolution, dynamically changing databases while retaining real time display (Haus, Newton, Ribarsky, Faust, & Hodges, 1996). It renders 3-D "realistic" terrain from an immense database of terrain data. This requires a significant computational and bandwidth overhead. Although a high degree of terrain realism can be achieved in VGIS, no 3-D symbology and model libraries are available.

The need is well recognized in the cognitive psychology literature (Barnes, 1997; Bennett, Toms, & Woods, 1993; Modrick, 1976; Paquet, 1992) for displays that are *process centered* and provide innovative visualizations and symbolic content. We intend to extend these cognitive engineering principles into the realm of 3-D real-time animated military planning. Our work attempts to meet the following desiderata for process-centered displays postulated by Barnes (1997): (a) develop objects that indicate the state of the events being displayed; (b) capture behaviors and rules of behavior; (c) represent possible end states for current battle trends; (d) represent process, goal, and environmental indicators; and (e) provide a means of executing and assessing various war-gaming scenarios.

We now describe the underlying system software architecture, recent improvements, and the continuous process of upgrading the software to enhance the ability of ABATIS to incorporate intelligent modules as visualization drivers. We show a realization of a war-gaming scenario fashioned after FOX-GA[2] (Schlabach, Hayes, & Goldberg, 1999), a genetic algorithm developed at the University of Illinois.

4. Software Development

Existing battlefield visualization systems typically exhibit high resolution and high realism. Their drawback is the lack of flexibility in modifying the symbology and war-gaming scenarios as well as the high overhead associated with the communication bandwidth that they require, especially when these systems are exercised in the intensely collaborative setting where such activities take place. The awareness of the tactical situation does not require all the details that such systems attempt to capture.

A number of themes should underlie any new architecture for battlespace visualization. Most importantly, the architecture must facilitate understanding of the *process* of the battle, rather than simply the current location of various forces. This requirement implies that the system should reflect how the user assimilates battlespace state information into a process-centered viewpoint. One aspect of this problem is the assembling of individual units of information into context-

[2] not an acronym

rich, higher level composites. Another is the presentation of this derived information in a way that is intuitive to the human user.

Motivated by these desiderata, the developers created the ABATIS system. Our key concept in the design of ABATIS is the process-centered display (PCD), a construct that can display complex, evolutionary processes as well as simple state changes (Keane, Rozenblit, & Barnes, 1997).

The main goal of the PCD's design is to convey the *processes* that are occurring in the battlespace. Since battlespace processes (e.g., maneuver, attack) evolve and change as the battle unfolds, the architecture must also support dynamic change and evolution at "run" time. Given the vast range of possible battlespace scenarios and objects, the architecture must also be flexible enough to permit the quick creation of new battlespace objects from old ones.

A secondary goal is to focus on the possibility of using motion, color changes, "morphing," or other types of animation to convey information. Some uses of animation are obvious, such as moving a symbol from one location to another. However, abstract quantities can also be tied to motion. A simple example would be allowing the strength of a ground force to be represented by the speed of rotation of its symbol. When done in a way that matches the intuitive notions of the user, such a presentation of information becomes a *metaphor*. The metaphor correlates familiar experiences with the actions of symbols on the computer display.

A final goal is to allow arbitrary levels of complexity in both the battlespace objects and their associated process dynamics. This complexity is needed to accurately model the intricate dynamics of a real battlespace and its metaphorical representation. Driven by these goals, the architecture for the ABATIS-PCD is designed using the object-oriented software design paradigm.

The fundamental design concept of ABATIS is the modularity of display elements. Terrain and unit elements are represented by symbols (objects) that can reside in libraries and can be placed on the display at any location and in any orientation. As opposed to the traditional paradigm of incorporating attributes and methods in object descriptions, we specify the behavior of such elements as distinct, generic entities that can be associated with the battlespace elements.

The process-centered display requires simple, fundamental classes from which instances of battlespace representations of any complexity can be rapidly constructed. More specifically, such classes are terrain, unit, behavior, and information (attributes). Unit objects can be built from elementary graphical elements (GRELS) (e.g., to construct a 3-D battalion symbol, we can use a rectangle, diagonals, and two vertical bars). New elements (with a more complex structure) can be created from the existing elements and can be stored in libraries. Thus, re-use and rapid construction of battlespace instances are facilitated.

The prototype of the ABATIS design has been implemented on the Silicon Graphics Octane machine, in the C++ programming language, using the Open Inventor™ development environment. The system's major capabilities are that it can

1. Load terrain elements, military units, and tactics into a scenario creation area.

2. Import any 3-D model specified in the Open Inventor™ format.

3. Construct objects from fundamental elements in the object creation window.

4. Replace a terrain or unit fundamental element.

5. Transfer fundamental elements to the scenario window to location (x, y, z).

6. Dynamically specify length and width of terrain size and scale objects and grid size.

7. Attach a behavior to a fundamental element in a scenario window.

8. Animate objects individually or synchronously.

9. Move an object according to a route by specifying a corresponding path name.

10. Dynamically alter global simulation speed for synchronous motion.

11. Execute a battle scenario by invoking war-gaming logic and assess it through the notion of configural displays.

5. Development of the Symbolic Battlespace Visualization Framework

Effective battlespace visualization should portray information in a way that gives a user the ability to intuitively understand the state of the battle (Barnes, 1997; Haber & McNabb, 1990; Hancock, 1994; Lehner & Adelman, 1988). We conceptualize, maneuver, and interact daily in a 3-D world. Thus, it is intuitive to visualize the battlespace in 3-D and demonstrate this intuition by creating realistic scenarios and empirically measuring combat performance as an integral part of the ABATIS architecture.

One of the most significant benefits of 3-D visualization is the ability to view graphical representations of objects from any perspective. Having the capability to visualize a 3-D object from any angle (length, width, or height) enhances the understanding of its characteristics. For example, in joint task force planning, it is necessary to provide a means of depicting air corridors and altitude. These are just two simple characteristics that 2-D representations lack. Thus, we anticipate that semantically rich designs of 3-D abstract symbology will allow the commanding officers to understand the battlespace more effectively. We envision an incremental development of concepts, based on the existing notational standard as well as on research into new ways of information portrayal. Consider, for example, the traditional mechanized battalion symbol and its evolution into the 3-D representation shown in Figure 1.

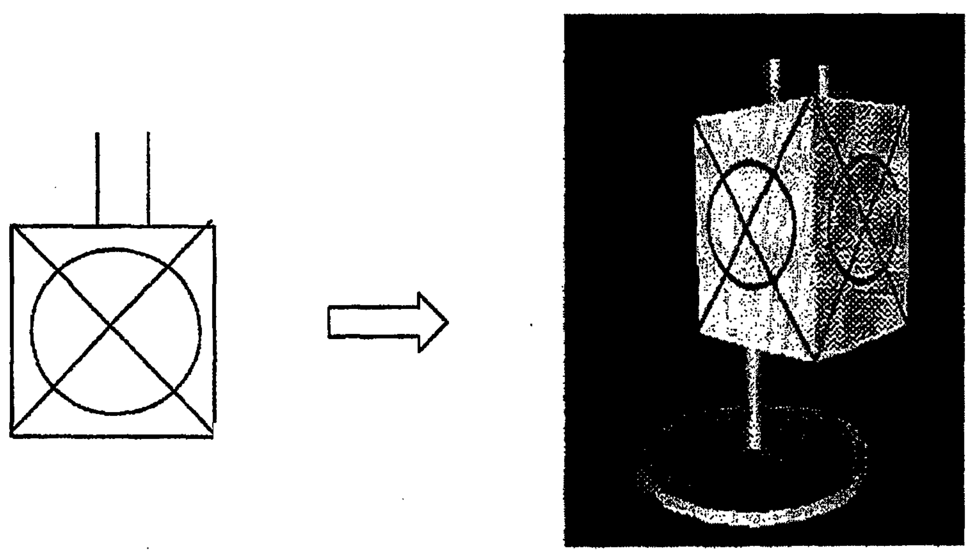


Figure 1. Evolution From 2-D to 3-D Symbology.

The 2-D symbolic representation of a battalion is composed of a square, two vertical lines, an oval, and an x-shaped symbol. The oval and the x-shaped symbol are placed inside the square to denote a mechanized unit, while the two vertical lines are placed on top of the square to depict a battalion unit. To translate the 2-D symbol into 3-D, the square is converted into a fundamental cube element, with the appropriate oval and x-shaped texture to denote a mechanized battalion unit. The two vertical lines are transformed into two small 3-D pipes. To give the battalion unit a height dimension, a flat cylinder and small pipe are used. The 3-D symbol is comprised of five separate elements (i.e., the "footprint," the stem, the cube, and two pipes), each of which could be ascribed behavior. It is a semantically rich vehicle for information representation.

For example, the footprint could metamorphose to a real ground trace via navigational data. The stem could show actual command post (CP) locations and could be used as a barometer display for supply status. The surfaces of the cube may be used to abstract various types of diverse information (e.g., Side 1 = the strength of the force; Side 2 = estimated time to destination, etc.).

Attributes can be attached to fundamental elements to signify a particular property. For example, colors can identify the affiliation of an element via the military's standard coloring scheme. Other properties can be expressed by the available graphical elements (e.g., a wire frame representation may indicate that the object is dead).

In addition to the fundamental unit symbology, we have refined terrain rendering. Rather than relying on computationally demanding digitalizations that require considerable storage resources, we provide 3-D abstractions of terrain elements that can be used to compose the basic terrain for military scenarios.

The abstract symbology with its relevant behaviors is used to provide commanders with decision support tools such as dynamic scenario generation and synchronous battlespace animation. The dynamic scenario generation is simple and rapid. First, the terrain is composed in the scenario window. Once the terrain is established, the user can place military units (both friendly and enemy) at any location within the terrain of interest. A battle scenario can be specified interactively and enacted by synchronous battlespace animation.

6. Battlespace Scenario Execution and War Gaming: A Model-Based Approach

To afford decision support, our architecture and its process-centered display must be driven by battlespace process models capable of rapid enactment and execution of war-gaming and intelligence scenarios. Our long-term vision of the architecture is an integrated system that spans a spectrum of processing methods and underlying physical elements. This design vision is shown in Figure 2. The architecture given is for a complete system that is capable of processing raw data and being used to drive the process-centered display. The architecture is arranged into *levels of abstraction* and separated into physical and procedural layers.

The physical layers comprise

1. The Database: Intelligence data collected through various sources (e.g., imagery, human intelligence [HUMINIT], signal intelligence [SIGINT], etc.; these are "raw" data).

2. The Battlespace Object Clusters: A collection of battlespace objects abstracted through the process of intelligence production.

3. The Metaphor Object Base: Metaphors are model engines that embody procedural mechanisms for displaying the battlespace state.

4. The Process-Centered Display: The procedural layers of the architecture would enable the transitions through the physical levels. Through intelligence production, data could be clustered, categorized, and amalgamated into objects that will eventually underlie the metaphors. Knowledge abstraction and mapping procedures will facilitate this process (i.e., they will provide mechanisms that should associate metaphors with the battlespace object clusters). The visualization and process dynamics control is a set of procedures and rules governing the change of graphical element states on the PCD.

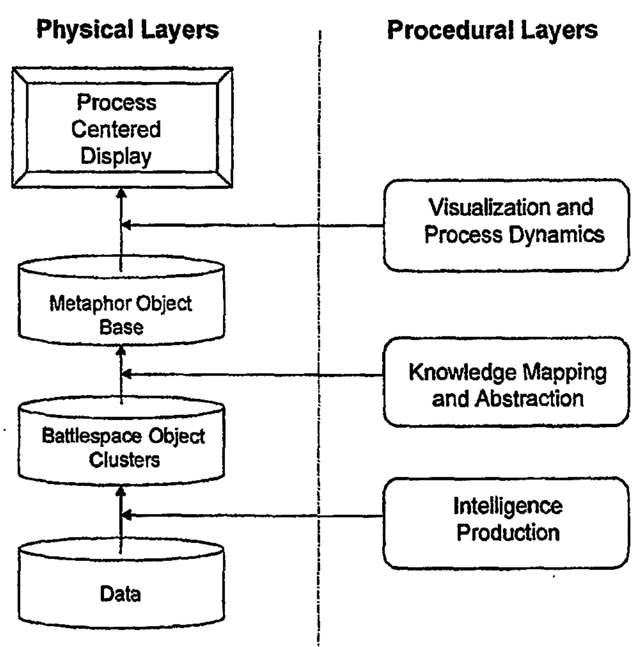


Figure 2. Integrated Battlespace System Architecture.

The procedural and physical layers are organized as modular objects that communicate by sending messages. The source of those messages can be a simulator or some other existing military software system adapted to that function. This modularity is intended to enable the PCD to "plug and play" with other commanders' decision support systems.

The three lowest physical layers are the basis for the construction of a model base intended to dynamically control the PCD. The lowest level is the raw data as they are acquired from the battlespace. These data may have many different formats and may be valid at various times in the past. For example, some data may be current, while other data may have come from sources that may be an hour old. Data at this level are relatively unorganized and unstructured.

Through the procedural application of intelligence production, the raw data are clustered or processed in some other way to produce the first level of abstraction. Battlespace object clusters are more closely related to the types of objects that commanders consider when they make tactical decisions. If a conventional user interface were applied to this level of the model, a display showing battlefield state but not battlespace processes would result.

Our approach to war gaming is based on COA generation and assessment concepts by Schlabach, Hayes, and Goldberg (1999). War gaming is the assessment of how well a specific friendly COA might perform in a battle against the enemy's COA (Kaiser & Proffitt, 1992). Therefore, as pointed out by Schlabach (Modrick, 1976), efficient COA generators and evaluators are critically needed tools that can assist the commander's decision making. The two COA generators, AirLand Battle Management and Systems for Operations Crisis Action Planning, do not facilitate assessments of how well the generated COAs would perform versus the enemy's COAs. The FOX-GA genetic algorithm-based COA generator and war gamer provides such capabilities. It uses causal reasoning to war game COA in a variety of scenarios. We plan to employ this generator in our system as the foundation for dynamic scenario execution. The FOX algorithm can provide us with the best COA and war-gaming rules, based upon which simulation model that drives the process-centered display can be built.

ABATIS is well positioned to interface with a war gamer such as FOX-GA. Figure 3 illustrates the modular design that facilitates integration with war-gaming rule bases and terrain and COA databases. Procedures that abstract those databases from a war gamer can be added.

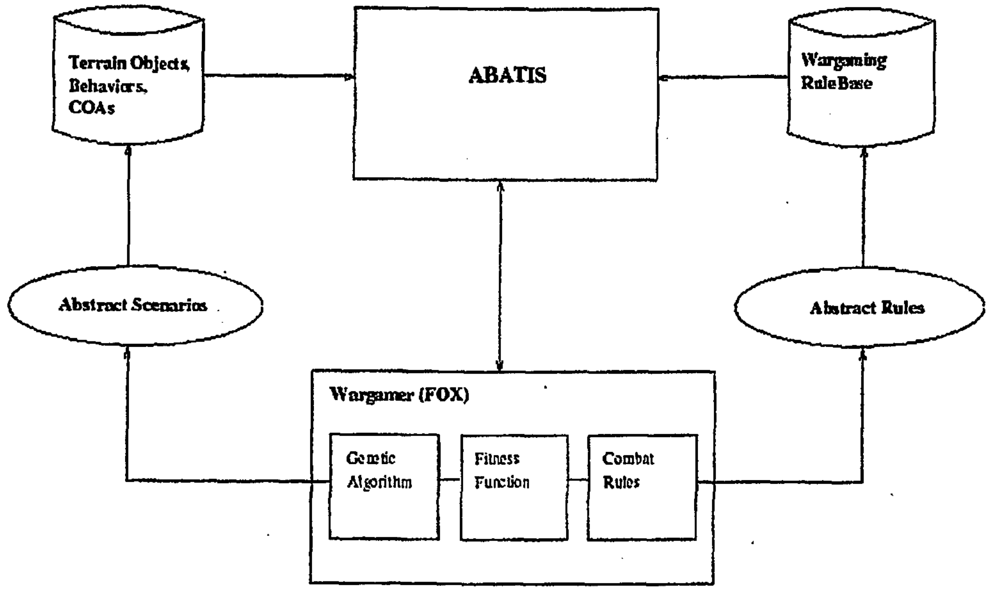


Figure 3. Integration of ABATIS With a War Gamer.

As a proof of concept, we have developed initial integration procedures wherein sample war-gaming logic abstracted from FOX is realized in an illustrative scenario.

7. An Illustrative Scenario

The scenario developed to support this research is a combined arms brigade executing a movement to contact mission. It is rather simple and straightforward in its implementation to allow rapid prototype demonstration of essential scenario dynamics rather than displaying high density, high resolution battlespace data.

7.1 The Area of Operations

The National Training Center (NTC), Fort Irwin, California, provides the geographic and operational setting for this brigade operation. The area of operations has two avenues of approach able to support multiple battalion formations. Viewing each from the vantage point of the line of departure (LD), one avenue of approach on the left allows virtually unrestricted maneuver. The other includes a significant choke point beyond the LD. Maneuver corridors for each avenue depict their relative ability to support mobility. There are three lines of defendable terrain (LDTs) beyond the LD. Friendly unit phase lines correspond with the LDTs. Each supports reasonable defensive operations by opposing forces, but there is no dominant key terrain favoring the defense. The nature of the terrain and the mission results in the designation of two battalion sectors. One corresponds to each maneuver corridor. The battlespace representation uses abstractions of terrain and man-made features. Certain terrain features that might appear on a standard military map are not included because they are not militarily significant. The terrain representation is kept austere because it will be evaluated for its adaptivity and the utility of its terrain information content.

7.2 The Friendly Maneuver Force

The brigade includes four battalion task force maneuver elements. Initially, they are positioned in assembly areas behind the LD. The left avenue of approach is designated as the brigade main avenue of approach. The right is the supporting avenue. Two battalions are in the lead echelon on the left with a reserve unit following in sector. One battalion is assigned to the right sector.

7.3 The Opposing Maneuver Force

The opposing force is defending lightly with a platoon-sized reconnaissance element at the choke point in the right battalion sector. In the left sector, two opposing force companies are arrayed along the second LDT. Two companies and residual forces from earlier positions in the sector will defend their main defense, which corresponds with the primary objective of the friendly forces.

7.4 The War-gaming Logic

The war-gaming logic is simple and fundamentally doctrinal. It is implemented to permit activation of basic battlespace dynamics and to demonstrate the responsiveness of the system to such logic. Attackers are favored whenever their combat power meets or exceeds 3:1. Combat power is calculated on platoon counts, not individual weapons or crews. Movement is controlled at approximately 5 kph when troops are not engaged and 0.5 kph when they are engaged. Specific attrition is keyed to three levels of relative combat ratios. Reduction of available forces below 65% triggers rearward movement or commitment of a reserve, when available.

7.5 Scenario Execution

The demonstration scenario flows smoothly from construction of the operating environment through friendly unit seizure of the objective. The abstract features provide excellent awareness of the tactical situation. A static instance of the scenario (excerpted from the ABATIS process-centered display) is shown in Figure 4.

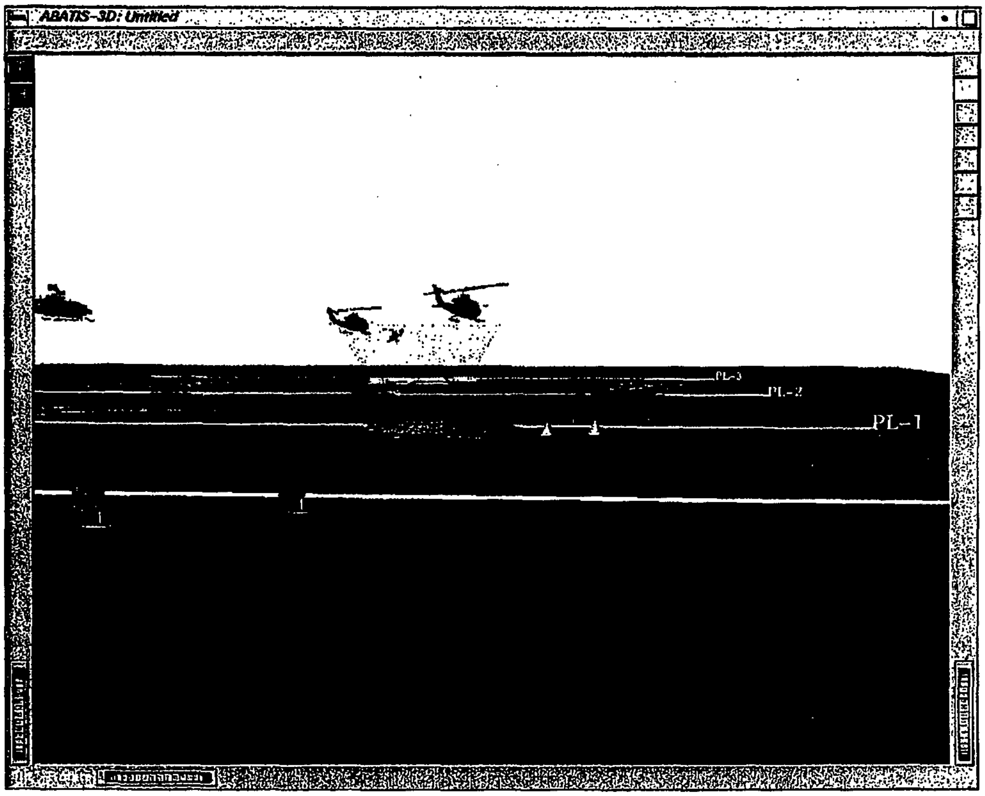


Figure 4. Instance of the Sample Scenario.

8. Summary and Research Issues

Based on the literature, we concluded that distrust of automated and decision support systems was a ubiquitous problem. Interestingly, we also found evidence that complacency and over-reliance on computer solutions stemmed from the same generic problem: lack of understanding of precisely what the computer is doing. These insights prompted a general research strategy to better understand the cognitive dimensions of using visualization as an interface between human and computerized problem solving. If the user understands and interacts with computerized solutions, then he or she can suggest, contradict, and if necessary, override computerized solutions. For this to occur, there has to be a common semantic framework between human and computer (a means of discourse) before any real synergy is possible. ABATIS is a software environment being developed to accomplish this by generating visualization concepts that will create a common semantic framework to forge efficient human-computer collaboration.

A number of important human performance issues must be resolved to expedite the semantic interface. The two identified as particularly important are effects attributable to the display of probabilistic information and effects attributable to cognitive biases, particularly, the confirmation bias. The working hypothesis is that better visualization methods will lessen the human limitations revealed in the literature. Better understanding of collaborative human-computer problem-solving characteristics will result in a semantic visualization environment that enhances dialogue between these two cognitive entities. The ABATIS environment will be the focus of our effort to understand this dialogue and to develop both principles and visualization concepts that will make future planning and re-planning a faster, easier, and more effective process.

References

Adelman, L., & Bresnick, T. (1992). Examining the effects of information sequence on Patriot air defense officer's judgments. Organizational Behavior and Human Decision Processes, 53, 204-228.

Adelman, L., Bresnick, T., Black, P.K., Marvin, F.F., & Sak, S.G. (1996). Research with Patriot air defense officers: Examining information order effects. Human Factors, 38(2), 250-261.

Banks, R., & Wickens, C.D. (1999). Commander's display of terrain information: Manipulations of display dimensionality and frame of reference to support battlefield visualization (ARL-CR-445). Aberdeen Proving Ground, MD: U.S. Army Research Laboratory.

Barfield, W., & Rosenberg, C. (1995). Judgments of azimuth and elevation as a function of monoscopic and binocular depth cues using a perspective display. Human Factors, 37(1), 173-181.

Barnes, M.J. (1979). Operator decision making characteristics (Technical Publication 6124). China Lake, CA: Naval Weapons Center.

Barnes, M.J. (1997). Process centered displays and cognitive models for command applications. Proceedings of the IEEE International Conference and Workshop on Engineering of Computer-Based Systems (pp. 129-135). Monterey, CA.

Barnes, M.J., & Wickens, C.D. (1998). Battlespace visualization: A multi-view approach. Proceedings of the 2nd Federated Laboratory Symposium (pp. 1-5). College Park, MD.

Bennett, K.B., Toms, M.L., & Woods, D.D. (1993). Emergent features and graphical elements: Designing more effective configural displays. Human Factors, 35(1), 71-97

Charles River Analytics (1998). Briefing on intelligent information management algorithms. Cambridge MA: Author.

Einhorn, H.J., & Hogarth, R.M. (1981). Behavioral decision theory. Annual Review of Psychology, 32, 53-88.

Endsley, M.R. (1995). Towards a theory of situation awareness in dynamic systems. Human Factors, 37(1), 32-64.

Graves, K., & Miller, D. (1998). Commander's intelligent battlefield information display (CIBID). SBIR briefing by CHI Systems, Inc., at the University of Arizona.

Haber, R.B., & McNabb, D.A. (1990). Visualization in scientific computing. In G. M. Nielson & B. Shriver (Eds.), Visualization idioms: A conceptual model for scientific visualization systems (pp. 74-93). Los Alamitos, CA: IEEE Computer Society Press.

Hancock, W.R. (1994). Meeting the graphical needs of the electronic battlefield. Proceedings of the 13th Digital Avionics Systems Conference, AIAA/IEEE (pp. 465-470).

Haus, J., Newton, G., Ribarsky, W., Faust, N., & Hodges, L. (1996). 4D symbology for sensing and simulation. Proceedings of the International Society for Optical Engineering. Orlando, FL.

Hollands, J., & Wickens, C.D. (1999). Engineering psychology and human performance (3rd ed.). New York: Prentice Hall.

Jones, P.M., Wilkins, D.C., Bargar, R., Sniezek, J., Asaro, P., Kesseler, D., Lucenti, M., Choi, I., Chernyshenko, O., Hayes, C.C., Tu, N., Liang, M., & Schlabach, J. (1999). CoRAVEN: Modeling and design of a multimedia intelligent infrastructure for collaborative intelligence analysis. Proceeding of the 3rd Federated Laboratory Symposium (pp. 3-7). College Park, MD.

Kahneman, D., & Tversky, A. (1973). On the psychology of prediction. Psychological Review, 80, 237-251.

Kaiser, M.K., & Proffitt, D.R. (1992). Using the stereokinetic effect to convey depth: Computationally efficient depth-from-motion displays. Human Factors, 34(5), 571-581.

Karr, C.R., Reece, D., & Franceschini, R. (1997). Synthetic soldiers. IEEE Spectrum, 39-45.

Keane, J.S., Rozenblit, J.W., & Barnes, M.J. (1997). The advanced battlefield architecture for tactical information selection (ABATIS). Proceedings of the IEEE International Conference and Workshop on Engineering of Computer-Based Systems (pp. 228-237). Monterey, CA.

Koehler, J.J. (1993). The normative status of base rates in jury trials. In N. J. Castellan, Jr., (Ed.), Individual and Group Decision Making (137-149). Hillsdale, NJ: Erlbaum.

Lehner, P.E., & Adelman, L. (1988). Senior battle staff decision aiding. A case study. In S. J. Andriole & G. W. Hopple (Eds.), Defense applications of artificial intelligence (pp. 153-166). Lexington, MA: Lexington Books.

March, J.G. (1978). Bounded rationality, ambiguity and the engineering of choice. Bell Journal of Economics, 587-608.

Marshak, W.P., Winkler, R., Fiebig, C., Stein, R., & Khakshour, A. (1999). Evaluating intelligent aiding in course of action decisions using Fox genetic algorithm in 2-D and 3-D interfaces. Proceedings of the 3rd Federated Laboratory Symposium (pp. 27-31). College Park, MD.

McCormick, E.P., Christopher, C.D., Banks, R. & Yeh, M. (1998). Frame of reference effects on scientific visualization subtasks. Human Factors, 40(3), 443-451.

McPherson, P. (1999). The character of Ulysses S. Grant. New York Review of Books.

Merwin, D.H., O'Brian, J.V., & Wickens, C.D. (1997). Perspective and coplanar representation of air traffic: Implication for conflict and weather avoidance. Proceedings of the 9th International Symposium on Aviation Psychology (pp. 22-86).

Modrick, J.A. (1976). Decision support in a battlefield environment. In C. P. Tsokos & R. M. Thrall (Eds.), Decision information (pp. 457-471). New York: Academic Press.

Mon-Williams, M., & Wann, J.P. (1998). Binocular virtual reality displays: When problems do and don't occur. Human Factors, 40(1), 42-49.

Paquet, L. (1992). Global and local processing in non-attended objects: A failure to induce local processing dominance. Journal of Experimental Psychology: Human Perception and Performance, 18(2).

Parasuraman, R., & Riley, V. (1997). Humans and automation: Use, misuse, disuse and abuse. Human Factors, 39(2), 230-253.

Patterson, R., Moe, L., & Hewitt, T. (1992). Factors that affect depth perception in stereoscopic displays. Human Factors, 34(6), 655-667.

Schlabach, J.L., Hayes, C., & Goldberg, D. (1999). SHAKA-GA: A genetic algorithm for generating and analyzing battlefield courses of action. Evolutionary Computation.

Schrieber, B.T., Wickens, C.D., Goetz, J.R., Alton, J.D., & Hickox, J.C. (1998). Navigational checking using 3-D maps: The influence of elevation angle, azimuth, and foreshortening. Human Factors, 40(2), 209-223.

Sniezek, J., & Buckley, P. (1993). Becoming more or less uncertain. In N.J. Castellan, Jr., (Ed.), Individual and group decision making (pp. 81-107). Hillsdale NJ: Erlbaum.

Sniezek, J., & Chernyshenko, O. (1998). Investigation of subjective probability estimates of military intelligence officers. Unpublished manuscript, University of Illinois, Champaign, IL.

Sollenberger, R.L., & Milgram, P. (1993). Effects of steroscopic and rotational displays in a three-dimensional path-tracing task. Human Factors, 35(3), 483-499.

Wickens, C.D., Merwin, D.H., & Lin, E.L. (1994). Implications of graphics enhancements for the visualization of scientific data: Dimensional integrality, stereopsis, motion, and mesh. Human Factors, 36(1), 44-61.

Wickens, C.D., Thomas, L., Merlo, J., & Sehchang, H. (1999). Immersion and battlefield visualization: Does it influence cognitive tunneling. Proceedings of the 3rd Federated Laboratory Symposium (pp. 111-115). College Park, MD.

Yeh, Y.Y., & Silverstein, L.D. (1992). Spatial judgments with monoscopic and stereoscopic presentations of perspective displays. Human Factors, 34(5), 583-600.

NO. OF COPIES	ORGANIZATION
1	ADMINISTRATOR DEFENSE TECHNICAL INFO CTR ATTN DTIC OCA 8725 JOHN J KINGMAN RD STE 0944 FT BELVOIR VA 22060-6218
1	DIRECTOR US ARMY RSCH LABORATORY ATTN AMSRL CS AS REC MGMT 2800 POWDER MILL RD ADELPHI MD 20783-1197
1	DIRECTOR US ARMY RSCH LABORATORY ATTN AMSRL CI LL TECH LIB 2800 POWDER MILL RD ADELPHI MD 207830-1197
1	OUSD(A)/DDDR&E(R&A)/E&LS PENTAGON ROOM 3D129 WASHINGTON DC 20301-3080
1	CODE 1142PS OFFICE OF NAVAL RESEARCH 800 N QUINCY STREET ARLINGTON VA 22217-5000
1	DPY COMMANDING GENERAL ATTN EXS (Q) MARINE CORPS RD&A CMD QUANTICO VA 22134
1	HEADQUARTERS USATRADOC ATTN ATCD SP FORT MONROE VA 23651
1	AIR FORCE FLIGHT DYNAMICS LAB ATTN AFWAL/FIES/SURVIAC WRIGHT PATTERSON AFB OH 45433
1	ARI FIELD UNIT FORT KNOX BUILDING 2423 PERI IK FORT KNOX KY 40121-5620
1	STRICOM 12350 RESEARCH PARKWAY ORLANDO FL 32826-3276

NO. OF COPIES	ORGANIZATION
1	US ARMY ATTN AVA GEDDES MS YA:219-1 MOFFETT FIELD CA 94035-1000
1	DR NORMAN BADLER DEPT OF COMPUTER & INFORMATION SCIENCE UNIV OF PENNSYLVANIA PHILADELPHIA PA 19104-6389
1	HUMAN FACTORS ENG PROGRAM DEPT OF BIOMEDICAL ENG COLLEGE OF ENGINEERING & COMPUTER SCIENCE WRIGHT STATE UNIV DAYTON OH 45435
1	COMMANDER MARINE CORPS SYSTEMS CMD ATTN CBGT QUANTICO VA 22134-5080
1	HQ III CORPS & FT HOOD OFFICE OF THE SCIENCE ADVISER ATTN AFZF CS SA FORT HOOD TX 76544-5056
1	MS DIANE UNGVARSKY HHC 2BDE 1AD UNIT 23704 APO AE 09034
1	ENGINEERING PSYCH LAB DEPT OF BEHAVIORAL SCIENCES & LEADERSHIP BUILDING 601 ROOM 281 US MILITARY ACADEMY WEST POINT NY 10996-1784
	DR SEHCHANG HAH WM J HUGHES TECH CTR FAA NAS HUMAN FACTORS BR ACT-530 BLDG 28 ATLANTIC CITY INTNATL AIRPORT NJ 08405

NO. OF COPIES	ORGANIZATION
1	US MILITARY ACADEMY MATH SCIENCES CTR OF EXCELLENCE DEPT OF MATH SCIENCES ATTN MDN A MAJ M D PHILLIPS THAYER HALL WEST POINT NY 10996-1786
1	ARL HRED AVNC FIELD ELEMENT ATTN AMSRL HR MJ (R ARMSTRONG) PO BOX 620716 BLDG 514 FT RUCKER AL 36362-0716
1	ARL HRED AMCOM FIELD ELMT ATTN AMSRL HR MI (D FRANCIS) BUILDING 5678 ROOM S13 REDSTONE ARSENAL AL 35898-5000
1	ARL HRED AMCOM FLD ELEMENT ATTN ATTN AMSRL HR MO (T COOK) BLDG 5400 RM C242 REDSTONE ARS AL 35898-7290
1	ARL HRED USAADASCH FLD ELMT ATTN AMSRL HR ME (K REYNOLDS) ATTN ATSA CD 5800 CARTER ROAD FORT BLISS TX 79916-3802
1	ARL HRED ARDEC FIELD ELMT ATTN AMSRL HR MG (R SPINE) BUILDING 333 PICATINNY ARSENAL NJ 07806-5000
1	ARL HRED ARMC FIELD ELEMENT ATTN AMSRL HR MH (C BIRD) BLDG 1002 ROOM 206B FT KNOX KY 40121
1	ARL HRED CECOM FIELD ELMT ATTN AMSRL HR ML (J MARTIN) MYER CENTER RM 2D311 FT MONMOUTH NJ 07703-5630
1	ARL HRED FT BELVOIR FIELD ELMT ATTN AMSRL HR MK (P SCHOOL) 10170 BEACH ROAD ROOM 12 FORT BELVOIR VA 22060-5800

NO. OF COPIES	ORGANIZATION
1	ARL HRED FT HOOD FIELD ELMT ATTN AMSRL HR MV HQ USAOTC (E SMOOTZ) 91012 STATION AVE ROOM 111 FT HOOD TX 76544-5073
15	ARL HRED FT HUACHUCA FLD ELMT ATTN AMSRL HR MY (M BARNES) GREELY HALL BLDG 61801 RM 2631 FT HUACHUCA AZ 85613-5000
1	ARL HRED FLW FIELD ELMT ATTN AMSRL HR MZ (A DAVISON) 3200 ENGINEER LOOP STE 166 FT LEONARD WOOD MO 65473-8929
1	ARL HRED NATICK FIELD ELMT ATTN AMSRL HR MQ M (R FLETCHER) NATICK SOLDIER CTR BLDG 3 RM 341 AMSSB RSS E NATICK MA 01760-5020
1	ARL HRED OPTEC FIELD ELMT ATTN AMSRL HR MR (M HOWELL) ATEC CSTE OM PARK CENTER IV RM 1040 4501 FORD AVENUE ALEXANDRIA VA 22302-1458
1	ARL HRED SC&FG FIELD ELMT ATTN AMSRL HR MS (C MANASCO) SIGNAL TOWERS RM 303A FT GORDON GA 30905-5233
1	ARL HRED STRICOM FLD ELMT ATTN AMSRL HR MT (A GALBAVY) 12350 RESEARCH PARKWAY ORLANDO FL 32826-3276
1	ARL HRED TACOM FLD ELMT ATTN AMSRL HR MU (M SINGAPORE) BLDG 200A 2ND FLOOR WARREN MI 48397-5000
1	ARL HRED USAFAS FLD ELMT ATTN AMSRL HR MF (L PIERCE) BLDG 3040 RM 220 FORT SILL OK 73503-5600

NO. OF COPIES	ORGANIZATION
1	ARL HRED USAIC FLD ELMT ATTN AMSRL HR MW (E REDDEN) BLDG 4 ROOM 332 FT BENNING GA 31905-5400
1	ARL HRED USASOC FLD ELMT ATTN AMSRL HR MN (F MALKIN) HQ USASOC BLDG E2929 FORT BRAGG NC 28310-5000
1	ARL HRED HFID FLD ELMT ATTN AMSRL HR MP DR A KARRASCH C/O BATTLE CMD BATTLE LAB 415 SHERMAN AVE UNIT 3 FORT LEAVENWORTH KS 66027-2300
	ABERDEEN PROVING GROUND
2	DIRECTOR US ARMY RSCH LABORATORY ATTN AMSRL CI LP (TECH LIB) BLDG 305 APG AA
1	LIBRARY ARL BLDG 459 APG-AA
1	ARL HRED ECBC FLD ELMT ATTN AMSRL HR MM (R MCMAHON) BLDG 459 APG-AA
	ABSTRACT ONLY
1	DIRECTOR US ARMY RSCH LABORATORY ATTN AMSRL CS EA TP TECH PUB BR 2800 POWDER MILL RD ADELPHI MD 20783-1197

INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	August 2000	Final

4. TITLE AND SUBTITLE

Soldier Performance Course of Action (COA) Visualization Aids

5. FUNDING NUMBERS

AMS: 622716.H700011
PR: 1L162716AH70
PE: 6.27.16

Contract No. DAAD17-99-P-0203

6. AUTHOR(S)

Rozenblit, J.W. (Univ of AZ); Barnes, M.J. (ARL); Momen, F.; Quijada, J. A. (both of Univ of AZ); Fichtl, T. (Compass Foundation)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of Arizona
Electrical & Computer Engineering Dept
1230 E. Speedway Blvd.
Tucson, AZ 85721

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Research Laboratory
Human Research & Engineering Directorate
Aberdeen Proving Ground, MD 21005-5425

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

ARL-CR-302

11. SUPPLEMENTARY NOTES

The contracting officer's representative (COR) is Michael Barnes, U.S. Army Research Laboratory, ATTN: AMSRL-HR-MY, Greely Hall, Bldg 61801, Ft. Huachuca, AZ 85613-5000 (telephone 520-538-4704).

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The computer revolution has resulted in extending the possibilities of battlespace visualization to the brigade commander and below. However, mobility and bandwidth considerations require that the systems be efficient to reflect the realities of modern combat. The Advanced Battlespace Architecture for Tactical Information Selection (ABATIS) is being developed to be a rapid planning and re-planning experimental environment. ABATIS's object-oriented architecture has the advantage of being able to rapidly construct a three-dimensional battlespace that will accurately represent the essential planning components of a brigade and smaller division battle environment. The basic architecture has been extended to include war-gaming logic as part of the software design, and examples are given that pertain to specific military problems. This capability will allow ABATIS to realize fully the implications of battlespace visualization by creating a human-computer synergy that encourages both human and machine to generate and evaluate possible courses of action and their consequences. The human performance implications are discussed, and particular attention is directed toward research issues related to terrain visualization, automation, decision making, and cognitive biases.

14. SUBJECT TERMS

battlespace visualization; human decision making; terrain visualization; course of action analysis; intelligent systems

15. NUMBER OF PAGES

30

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

THIS PAGE INTENTIONALLY LEFT BLANK

DOCUMENT 2

An Intelligent Training Management System (ITMS)

AD-A373078

January 2000

Stottler Henke Associates, Inc.
San Mateo, CA

THIS PAGE INTENTIONALLY LEFT BLANK

An Intelligent Training Management System (ITMS)

Richard Stottler

Final Technical Report

January 28, 2000

DISTRIBUTION STATEMENT A: Approved for public release; distribution is unlimited.

Stottler Henke Associates, Inc. (SHAI)
1660 So. Amphlett Blvd., Suite 350
San Mateo, CA 94402
(650) 655-7242

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE January 28, 2000	3. REPORT TYPE AND DATES COVERED Final Report

4. TITLE AND SUBTITLE
An Intelligent Training Management System (ITMS)

5. FUNDING NUMBERS
F41624-99-C-6022

6. AUTHORS
Richard Stottler

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Stottler Henke Associates, Inc.
1660 South Amphlett Blvd., Suite 350
San Mateo, CA 94402

8. PERFORMING ORGANIZATION REPORT NUMBER
No. 194: ITMS Final Report

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory/AFRL/HEAI
Further Marked For: Terresa E. Jackson
2504 Gillingham Dr., Bldg. 170, Suite 25
Brooks AFB, TX. 78235-5100

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
NONE

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Report developed under SBIR contract for topic number AF99-097. Air Force training units are in extreme need of advanced, intelligent training management systems to aid the training managers and schedulers in the performance of their duties and to help students quickly advance in their careers and meet the training requirements. The intelligent training management system (ITMS), to be implemented and used in Phase II, will address tracking, evaluation, requirements identification, scheduling, and completion and certification management of individuals and teams. The ITMS will perform the functions that a person dedicated to managing the training of a small group of students would perform, but do it automatically. It will intelligently guide the students as to their training needs and opportunities and help with the development, delivery, scheduling, and evaluation of courses and other training events. The ITMS intelligently models the skills and knowledge mastered by the student and makes intelligent proactive decisions and notifications based on that model. It also provides intelligent courseware tracking, evaluation, and configuration control. After determining training requirements, it intelligently schedules the required resources. The ITMS is a general tool which can be easily customized to specific domains by end-users. Several sites will use the ITMS operationally in Phase II to provide feedback and a basis for follow-on commercialization.

14. SUBJECT TERMS
SBIR Report, Training Management, Intelligent Scheduling, Artificial Intelligence, Team Training,

Intelligent Decision-making, Data-Mining, Intelligent Training Management, Student Modeling

15. NUMBER OF PAGES
83

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNLIMITED

Table of Contents

1.0 Summary

In our Phase I project we accomplished all of the objectives listed in the Phase I proposal. Perhaps most important was the fact that we proved the feasibility of the concepts described throughout this final report in a prototype, which implemented functionality in most of the areas envisioned by the Phase II design included herein. For example, it had an E-mail interface and could send students proactive notifications through E-mail. It modeled the students general and specific skills using a skill hierarchy which could be edited by users and included both the more specific and subtask relationships. Users could edit the course descriptions, which included different versions for the same course. The course descriptions included prerequisite skills required and the skills developed (learning objectives) by the course along with minimum computer requirements.

The ITMS ran on a web server and wrote relevant information to the appropriate web pages. When a new version was created, a student who had taken the old version would be notified both through E-mail and through a web page tailored to the specific student. The prototype ITMS would notify students when they we're falling behind or when their skills had decayed. It would automatically E-mail them questionnaires after they took a course and expect a response in a reasonable amount of time. This time expired the student was reminded until eventually his supervisor was notified via E-mail.

Users could edit job descriptions and edit the career map, graphically specifying prerequisite relationships. This information was used to provide career counseling to students on their own web page. This included determining if the goals were realistic and determining what course the student needed to take and in what order. If the student's existing skills didn't meet those required as prerequisites for a course, the prototype would search for additional courses that the student was eligible for that would allow him to reach the required skill levels.

The prototype even had permission management capabilities. Students or instructors could be individually authorized with passwords and each had privileges appropriate to their class. The prototype would also evaluate courses based on the data that it had from students, supervisors, and the course results. This was a simple version of the algorithms described in Section 3.

The design for the phase II system, and therefore for the prototype, whose primarily role was to prove that design's feasibility, was based on interviews with operational experts in the training management process. Our first discussions were with various Air Force officers with AWACS team training management responsibilities at Tinker AFB in Oklahoma. We also received and analyzed documents relating to the complex training requirements relating to aircrew (especially pilot) training. We also had discussions with the several individuals with training management responsibilities at the Army's military intelligence distance learning center at Fort Huachuca, Arizona. Finally we confirmed that the Navy's needs paralleled the Air Force's and Army's through discussions with Commander Pinto, XO of the USS Paul Hamilton.

2.0 Problem Description

The Department of Defense' (DOD's) training requirements are changing, primarily because the jobs that DOD personnel do are changing. More thinking is required of all military personnel at all levels, primarily problem-solving – thinking through difficult problems. These changes are a result of "the new world order." That is, with the end of the cold war, the US faces asymmetric threats (enemies with far less capability than itself). Low intensity conflicts and military operations other than war (MOOTW) are more the norm than the exception. These lead to unpredictable situations and ill-structured problems. These circumstances require a higher degree of flexibility in the individuals.

Additionally, there is a much greater number of these asymmetric threats. During the Cold War we faced one, or possibly a few, credible threats with known doctrine. This could be studied and tactics developed, in advance, to counter likely enemy actions. But now, with hundreds or thousands of possible threats, many only vaguely known or even completely unknown, it is impossible to study or understand all the possible adversaries, their capabilities, doctrine, and tactics. Thus, it is impossible to design appropriate responses to their actions in advance and train our military personnel in those actions. Given the unknown nature and behavior of the threats, cognitive flexibility of our forces is paramount.

The world continues to rapidly evolve in other ways as well. Equipment rapidly changes – both those in use by our own forces and those used by potential threat forces. New software versions used by our forces may be updated multiple times per year. It is impractical to retrain our military personnel in the specific capabilities of a new equipment (either equipment of ours or possessed by potential threat forces) every time it changes. Rather our personnel need to have the general abilities to adapt to new equipment, without retraining each time. Their original training should not be for specific equipment, but rather how to understand the new capabilities, on their own.

Because the requirements of military jobs have changed, the training for those jobs must change. For example, there is already less emphasis on training specific procedures, preprogrammed reactions, doctrinal rules, rote memorization, and behaviorist training approaches. More emphasis is already being placed on training in the context of scenarios. That is, training is being conducted by practicing for a variety of situations. There are several theories of learning that relate to training with scenarios. These include Situated learning, Anchored instruction, Scenario-based training, Simulation-based learning, Case-Based instruction, constructivist theory, and several others. Many of these place explicit emphasis on the notion of training more abstract, general, problem solving skills and less emphasis on specific, concrete procedures and tasks. Because these training strategies embody more complex methods of instruction, and because there is a greater emphasis on more general (and subtle) skills, more complex tracking of student training is required. It is not enough to simply know which student took which course, because two different students may have had widely varying experience in the same course, and evaluating and tracking general problem skills requires more sophistication. It is not enough to simply track which course a student took or even which of his responses were correct and incorrect. Instead a system must evaluate and track his mastery of both specific and general skills and knowledge.

It should be noted that we have a very general definition for the term "course" as used herein. A course is any defined learning experience or training event and includes Distance Learning, correspondence courses, residence courses, on-site training, training sorties, simulator training, on-the-job training, etc.

The DOD training situation is also changing. Training budgets are being radically reduced. This is forcing training units to radically reduce or eliminate school house course lengths. This training is being replaced with distance learning alternatives, either as optional training or as explicit prerequisites for other courses, jobs, or promotions. The new courseware being produced to fill this void varies widely in terms of level of sophistication and pedagogy, depending on who is developing it.

The DOD also has very unique training (and training management) requirements. The training is often life-critical with complex, time constrained, high-value decision-making. The training requirements regulations are very complex. There is a need to provide just-in-time training which is specific to a particular mission or geography. Finally, the DOD has begun to appreciate the importance of managing the training of its personnel across their entire career – the Life Long Learning concept. This leads to very long student tracking times.

Many of these problems specific to the DOD are exasperated by the specific needs of Air Force aircrew team training. Air Force training units are in extreme need of advanced, intelligent training management systems. As warfare has gotten more complex (technology and tactics), Air Force training requirements have increased. But at the same time, Air Force training budgets have been reduced. This is true for both initial and sustainment training of active duty, guard, and reserve personnel. Training which includes academic schools, simulators, flight, on-the-job training and distance learning courseware, are all affected and in need of a new generation of training management systems.

The reduced budgets and increased requirements have forced training units to do more with fewer resources. Optimal utilization of these scarce resources has become more important. This has significantly increased the importance, complexity, and difficulty of scheduling training events. Even after a good schedule is developed, it is subject to many dynamic events and constant rescheduling is the norm. This occurs for several reasons including weather (not acceptable for training requirements), lost resources (equipment breaks or is re-allocated), trainees becoming unavailable, etc.

Determining the training requirements for individuals is also complex. This is because the regulations governing training are so complex and any one individual is subject to many different regulations. For example, some training, such as small arms training, is required of all Air Force Personnel. Other training requirements apply to all air crews, such as altitude chamber training. Additionally, pilots (who are also subject to the requirements for air crews and all personnel) must stay current on the relevant air platforms or they will require refresher training. As this example shows, any particular individual may be subject to many separate requirements. Furthermore, many of these regulations have complex periodicity rules. For example, a pilot may be required to log a certain number of flight hours each month, without too great a period of time transpiring between flights. Missing the month target may then activate the 90 day minimum requirements. Depending on the degree of the deficit, there will be different training requirements to make the pilot current again.

With each individual having to meet so many and such a complex array of requirements, tracking which requirements have actually been met is also difficult. Not every student attends every event for which he is scheduled. Thus, the "as-attended" record of the training events must be used.

These problems are further exacerbated for the many team training domains. An AWACS aircrew has about 18 positions. Thus, 18 different trainees have to be scheduled for most AWACS training events. Most of these fall into different categories, each with its own unique training requirements. Additionally, several different types of instructors have to be scheduled, who have their own training requirements to meet in order to be acceptable as instructors and to be allowed in the air. In the AWACS domain, non-AWACS aircraft, which are under control of different units altogether, must also be scheduled for most airborne training events. These aircraft may or may not have symmetric training requirements. For example, while tanker aircrews have training requirements relating to practicing with AWACS crews, and are thus relatively easy to negotiate schedules with, fighters have no such requirement. That is, while the AWACS Weapons Directors (WDs) have a training requirement to control fighters in various airborne scenarios, the fighters have no corresponding training requirement to be controlled. They can be very difficult to schedule.

These training management problems are especially difficult for reservist and guard units, whose members have full-time jobs. They are not available every day. Furthermore, on occasion, something happens at the trainee's job which prevents him from making his scheduled Air Force training event, sometimes with little or no notice.

We have spoken to several different individuals involved with the training management problems for various units who train out of Tinker AFB in Oklahoma. All were very informative and eager to help and provide many useful examples of current problems, some of which are described below.

An example from the reservist AWACS domain helps illustrate the training management problems. One of our contacts proposes that scheduling AWACS reservist sorties is one of the hardest training management tasks in the Air Force. First, he has to coordinate the individual schedules of the trainees manning the 18 separate positions, each of whom has another full-time job, which creates additional scheduling constraints. These jobs make the typical 8-12 hour active duty AWACS sortie length impractical. So the sortie length is generally reduced to 6 hours. Furthermore, the trainees are only available at certain times during the week. These problems are magnified by 6 since he is in charge of scheduling for 6 different teams. He then has to coordinate his training sorties with fighter training sorties in the area. The fighters fly in several different profiles, many of which generate no events, and therefore, no training for the AWACS crew. So he must piggy back his AWACS sortie onto fighter sorties who are scheduled to fly the correct types of missions. Finding these is a difficult problem since the fighters can more easily fulfill these certain profile training requirements using ground controllers endemic to their units.

Even after this complex schedule of sorties is worked out, problems often materialize. It is common for the fighters to cancel their sorties with 12 to 24 hours notice. In order to preserve the training sortie and reservists training schedules, the scheduling team then quickly scrambles to find alternate fighter training sorties. Usually they call units in a four to five hundred mile radius. They typically call fighter reserve units first. This is for two reasons. The fighter reserve units can be more understanding of the special needs and problems of other reservists. Secondly, since reservists tend to be more experienced than their active duty counterparts, they make fewer mistakes, so the fighter reservists prefer to train with AWACS reservists. About 75% of the time, they are successful in finding alternate reservist fighters to be controlled. The next fall back is to try to coordinate with active duty fighter training. The final fall back is to perform only surveillance training. That is, there are some training requirements for some of the AWACS crew that can be fulfilled by using the onboard sensors to monitor commercial air traffic. But certainly, weapons directors fulfill no requirements in this final fall back mode of training. Because of these problems and issues, maintaining the training sortie schedule for 6 AWACS crews requires 4-5 full time schedulers.

The most time-consuming aspect of their jobs relates to creating the schedule and disseminating it (and the frequent changes and updates) to the wide variety of individuals involved. In addition to the many people directly involved in the training (trainees, instructors, pilots, etc.), there are a large number of people not directly involved who also must be notified. For example, the maintenance unit, who is otherwise not involved in the training, must be notified so that they will be available and so that the equipment is available.

Another Tinker AFB contact was in charge of the training management for about 25 AWACS Weapons Director reservists. This requires about 30 hours per week. Her first problem is determining the training required for each individual. This is complicated by several factors. First, an Air Force wide automated system keeps track of each individual's flight hours. Unfortunately, there is a two step entry process, where the trainee fills out a form which must be entered by someone else. Occasionally, this is not done correctly, and the error is usually not uncovered for several months or longer. By that time, it is often difficult to remember or reconstruct which flights which trainees were on. Furthermore, as alluded to earlier, the regulations which describe what the training requirements are, are themselves complicated. Keeping track of how often different recurrent training must be done (whether quarterly, yearly, every 2 years, every 6 months, etc.) and matching those to each individual is difficult. Flying requirements are more complicated, requiring a certain number of days per month, without too long between flights and fall back requirements spanning 90 days. Again these must be applied to each individual. The result is a complicated array of training events for each trainee which includes such diverse items as life support training, chemical warfare instruction requiring special equipment, altitude chamber training, academic training, simulator training, training sorties, etc.

Once the events each trainee needs are determined, significant difficulties remain. Scheduling of training events occurs in a distributed manner. For example, the AWACS training manager (managing the training for 25 people) must coordinate with the training sortie manager (who provided the other example, above) as well as a large number of other individuals who each manage their own set of required training resources and events. A negotiation process occurs where they balance the needs and schedules of trainees against the availability and schedule of resources and related training events.

Even once the schedule is set, problems occur. The training manager discussed several types of problems. One problem, more common with reservists, is last minute cancellations, due to illness or job requirements. This requires rapid rescheduling to fill the required position, so that training can occur for the other positions, hopefully with someone who needs the training in that position. Furthermore, care must be taken to track the fact that the originally scheduled trainee did not attend the event and therefore still has an open requirement for it.

Another problem that occurred recently was that an instructor was unable to make the training flight. This resulted in a lot of last minute scrambling to find someone with the applicable training credentials to fill the spot. A system which keeps track of all training resources, including instructors, both as to their availability and their whereabouts, would greatly aid this rescheduling process.

A third contact reiterated many of the same problems and issues, but also brought up several others. It is important for senior leadership to see how far along the trainees are so that they can determine how many more sorties or how much more simulator (and other resource) time to buy. Furthermore, an automated system must be able to handle large training events and large aircrews. The system must provide Web access to allow trainees to sign up for courses and events and provide their availability information. An intelligent system must be capable of deconflicting the schedules of trainees and of the resources. One problem they currently have in particular is having trainees scheduled for two separate events at the same time.

Widely varying and rapidly changing training requirements result in there being many different versions of the same course. At a particular moment in time, there may be different versions of the course in terms of different scenarios (perhaps for different types of missions or different geographical locations) or for different types of computer hardware. Particular training organizations may need to frequently update their courses as well, leading to multiple course versions each year.

Although there are a very large number of existing training management systems, these do not begin to meet the complex needs discussed here and do not contain intelligent features. These systems are primarily networked database systems and store data relating to course catalogs, class schedules, enrollment, student information, transcripts, class evaluations, homework, self-assessments, course authoring, content management, grades/test scores, and rudimentary skills. The primary benefit they provide is that of a pre-customized DBMS with existing interfaces defined to the vendor's own courseware offerings or authoring tools. The primary disadvantages are that they do not attempt to track higher level skills and they do not exhibit intelligence, decision making, or proactivity, leaving these functions to the training managers or the students themselves.

An intelligent system is needed to help manage the complex training process. It should perform the functions that a person dedicated to managing the training of a small group of students would perform, but do it automatically. This would achieve an ideal which is rarely achieved. An Intelligent Training Management System (ITMS) would intelligently guide the students as to their training needs and opportunities and help with the development, delivery, evaluation and scheduling of courses.

The problems discussed above dictate the requirements of an ITMS. The ITMS will keep track of what general and specific skills, knowledge, and tasks the student has mastered over time. It will use that information to proactively help the student manage his career and the life-long learning process. After determining the training requirements, it will schedule the required training resources, including instructors and other team members (other students). It will track the different, changing versions of courses and help manage the change and notification process. For example, it will keep track of which student took which version of a course and know how they are different. Given relevant data, the ITMS will automatically produce an evaluation of each course. In addition to capabilities for students, instructors, and course authors, it will provide functionality for supervisors, mentors, course evaluators, and training managers. The ITMS will support the management of the permissions and authorizations to access the various data and functionalities.

The ITMS will provide intelligent student tracking and learning/career management. It will keep track of where the student is at in his career, in terms of what courses and jobs he's had. But more importantly, it will keep track of what skills, knowledge, and tasks he's mastered over time. These will include general and abstract skills, not just specific, concrete ones. For example, one new skill is the ability to learn about new equipment and how to troubleshoot it. Another is the ability to adapt to new enemy capabilities. When tracking a student over a long period of time, many things can change. His job requirements change. The courses change. Some of his skills decay from lack of use. (That is, skill mastery doesn't always increase). Because the ITMS is tracking these skills over a student's entire career, and because the required skills change frequently, the ITMS must allow the training managers to update the general and specific skills taught by courses and required by jobs. The ITMS must be able to track prerequisites taken by the student and required by him for future events. The ITMS will be able to calculate these prerequisites, even given the complexities of determining them in the face of complex regulations and skill requirements.

The ITMS will be proactive – telling students what prerequisites they need to finish before taking courses, nudging them when they fall behind, and informing them of possible skill decay. This will occur when the student is only using part of the skills for which he was trained, in his current job. This is especially important if his next job assignment will be using a different set of skills. In that case, it should evaluate those skills (with a no-penalty test) and remediate the student with refresher training as appropriate. The ITMS will inform students of the need for updated knowledge and skills for their jobs and new courses or new versions of old courses that address those deficits.

The ITMS must also address individuals and teams. The ITMS must be able to independently make decisions and recommendations but also accept input and overrides from training personnel. As part of determining the training requirements and tracking their completion, the ITMS could also evaluate the results of the various training events, either for individuals or teams. This would allow it to be able to automatically schedule additional, or remedial, training as required. Another issue which an ITMS could easily address is that of deployment. When a team must be quickly assembled to deploy, the ITMS has all of the skill, training and availability information to select the best team, either as a whole or assembled from individuals. It can also identify the additional training required of a team and its members to meet the needs of a particular deployment.

3.0 Solution Overview

The Intelligent Training Management System's primary focus is the student and its primary objectives are to maximize his efficient training and to further his career development in the context of life-long learning and general problem-solving. The tools it has available to it with which to accomplish these objectives are primarily the different types of learning opportunities and training events, and, evaluation methods, although all of these are constantly changing. These include distance learning, on-site, and correspondence courses; on-the-job-training; tests, just-in-time scenarios, simulator training, training sorties, etc. In order to make decisions regarding its actions, the ITMS has several types of knowledge available to it, including prerequisites, course learning objectives (which skills are taught by the course), training requirements regulations, job descriptions (which skills are required and practiced by various jobs), estimates of the decay rate for those skills, available resources, and the career map which describes the progression and prerequisite relationships between courses, ranks, and jobs. All of these change over time. The ITMS also has sources of additional knowledge including the student, his supervisor, course results, and evaluation results.

The ITMS will determine the applicable training requirements for trainees and teams. It will schedule the required events, including the trainees, instructors, and other needed resources. It will track the results and update the trainee's and team's histories. It will be able to select the best teams for deployment on particular missions and what training is required for a particular team to perform a particular mission.

The student tracking solution is based on the intelligent student model, borrowed from the realm of intelligent tutoring systems. The ITMS explicitly models the student's currently mastered skills, knowledge, and tasks. These are the stated and more general learning objectives of courses. They are organized by the instructor as a multiple dimensional hierarchy primarily around the more-general and subtask relationships. The student model also includes a description of which jobs the student has had and which courses he has taken. Since both are subject to change over time, the student model actually references specific versions of each. The courses and job descriptions utilize the same vocabulary (the hierarchy of skills, knowledge, and tasks) used by the student model. These are used to infer mastery levels in the student model. The ITMS will, when appropriate, automatically question the student and his supervisor regarding the specific and general skills taught by courses, and the degree to which they are successfully taught, skills required for (or not used in) various jobs, and the student's current degree of mastery of those skills.

The student's mastery changes either up or down over time. This is modeled with estimates of the skill increase provided by courses and jobs (on-the-job training or simply practice and experience) and heuristic skill decay factors which become specialized to the student, through data mining techniques, after the ITMS had had a chance to observe him over a long period of time. The ITMS also knows how quickly the student should be completing courses and progressing in his career. The ITMS uses the student model to proactively make decisions and notifications. It also contains more mundane information such as contact information (E-mail, phone, mailing address), available computer resources, his supervisor, etc.

After training requirements for each team and individual are determined and approved, the system would attempt to schedule the applicable events. The ITMS would make use of each student's availability, constraints, and requirements to come up with the most desirable schedule for the team as a whole. It would also have to negotiate with the managers of the training events or applicable resources. This negotiation might be with the human managers, in which case they would be sent an e-mail with a form to check-off the possible available dates and capacity of the required events. Or it might be with an ITMS component, which is local to the training event or resource manager. In that case, several messages can pass back and forth as to an optimal event schedule, given the needs of the students and of the events and resources. All related training managers could alter the schedule or add additional constraints. Since we have found in most scheduling problems that there is often a large number of reasonable schedules, and since not every constraint is always defined to the scheduler, the ability to manually adjust the schedule, while the ITMS continues to check for constraint violations or resource conflicts, is very useful.

The ITMS would provide reports to senior leadership about the progress of the training and the additional resources required to meet applicable training targets. When requested, it could assemble or select the most applicable team based on mission requirements. It could also determine the additional training that is required to bring one or a set of teams up to the requirements of some particular type of mission. If requested, it could then schedule the applicable training events.

Along with the student model is an intelligent course model. It uses the same vocabulary (skills, knowledge, and tasks hierarchy) as the student model to describe its learning objectives. Because a course will have different versions (such as which scenarios were actually used for a particular student as well as due to updated content over time), each course is actually a complex web of versions and scenarios. Each version has its own (partially different) learning objectives, history and student lists. The ITMS automatically uses actual student performance to evaluate the course in terms of how well it meets its learning objectives; that is, how well it teaches the specific and general skills.

The course model is used by the ITMS as its starting point for students who have taken the associated course. ITMS's first estimate of the mastery of a skill by a student is based on the results from the course that first teaches that skill to the student. (This estimate is later updated based on supervisor evaluations, relevant on the job experience (or lack thereof), decay factors, and future learning events). The course model can handle prerequisites in one of two ways. Courses, jobs, or ranks can be explicitly listed as prerequisites for other courses jobs, or ranks in a career map. Or, the required mastery level of skills, knowledge or tasks can be listed. This latter method provides more flexibility for the ITMS in terms of how it recommends that prerequisites be fulfilled. For example, if one course is listed as an explicit prerequisite for another, the ITMS will be forced to require the student to take the first course before the second. However, if a course lists skill levels as its prerequisites, there may be multiple methods of achieving those requirements. It is even possible that the ITMS estimates that the student already has the prerequisite levels (perhaps through on-the-job experience, or other courses).

The training manager will also be able to specify rules to allow waiving of prerequisites. The ITMS will calculate their effect in terms of the number of eligible students and likely course throughput.

Given the intelligent course model, it's relatively straightforward to add version control and tracking and configuration control functionality. Version control and tracking includes keeping track of which courses and versions are available and what each teaches; which students or units have which versions, whether they were distributed via CD or the Internet; making sure students are using the correct version for their particular needs given their computer constraints; and making sure the appropriate students are notified of course updates.

Configuration control refers to aiding the courseware development process by tracking the different versions of the separate files that make up the courseware, making sure that all the development team is using the most up-to-date versions, and that the final released product is the most up-to-date version. These capabilities are important when several different individuals are involved in authoring the course.

A final ITMS capability is automatic courseware evaluation. It is facilitated by the student and course models. The ITMS evaluates the course's ability to meet stated and more general learning objectives. It will use both in-course test results (which is somewhat circular) as well as after-course test results. The ITMS will use job performance, based on questioning the supervisor on specific and general skills of the student. It will also use the student's performance in follow-on classes. It will initially evaluate the course-based results from a pre-release test class, if available. It uses the information in the student models to estimate a course's ability to impart skills, knowledge, and task mastery. Using constraint satisfaction, it can also assess the course in reference to specific students and use data-mining techniques to look for patterns to see what attributes a student needs to lead to a good or bad performance in particular courses. This is helpful feedback for the course authors since it tells them if their course is particularly good or bad for certain kinds of students, so they can take advantage of it or take the opportunity to fix it.

4.0 System Description

4.1 ITMS Architecture

The high level ITMS Architecture is given below. The ITMS resides on a web server. We have identified 8 different types of uses, each of which interacts with the ITMS primarily through a web interface customized to that type of user. Each is described further below. The ITMS updates each user's specific web page with information, which is particular to that user. Additionally, for proactive notifications, the ITMS will be interfaced to an E-mail system so that it can send E-mail to any users that have it. The ITMS will also have the capability to print out physical letters, in the event that a user is unreachable with E-mail. The ITMS will also have a database interface to receive information from external databases and update them, if required. These databases include personnel databases (to get a student's contact information, current rank and job, supervisor, etc.), registrar databases (to determine who is currently registered in what course, what future courses, what past courses, and any certifications), course results databases (to get student's course results), and course description databases.

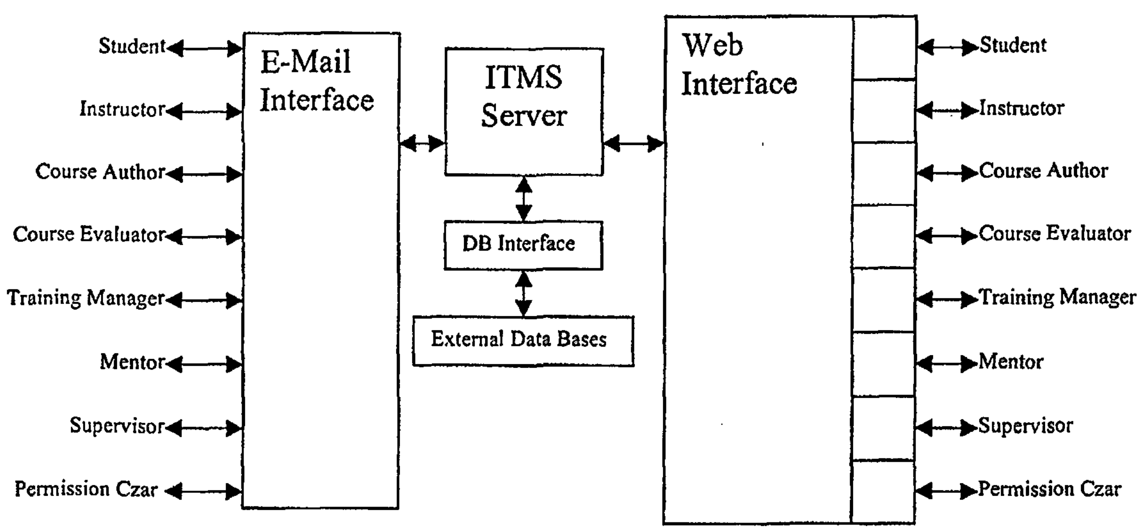


Figure 1. ITMS High Level Architecture

The student has a personalized web page that ITMS updates with information specific to him. For example, if there is a new version of a course that he has taken, that information will be posted on his web page. When the ITMS schedules him for a particular training event, that information will be E-mailed and posted to his personal web page. The student can use his web page to make queries and generally see what courses are available and what information the ITMS has deemed especially relevant to him. This is also where he keeps his contact and other personal information up-to-date, including his availability for training sorties or other hard-to-schedule events with limited resources and opportunities. If the ITMS has had trouble contacting him, it will specifically request up-to-date contact information when he logs on. The student can view his own skill mastery levels, and receive career counseling guidance as well. The proactive E-mail-type notifications include the need to take prerequisites, the fact that he is falling behind in completing those prerequisites or other requirements of his career path, updated versions of courses or knowledge required for his job, and skill decay warnings.

The instructor has authorization and capabilities through the web page to view the models of students currently in his courses, add new students to his courses, remove students from his courses, and register the student's result data for his courses. He can also view student questions or products and send answers to all the course's students, a particular student, or post them to the course web page.

The course author maintains the descriptive information for course versions through his web page. He can view the skill requirements and other prerequisites for various jobs, edit the skill and other prerequisites for his courses, edit the estimate of the specific and general skills mastery that the course accomplishes (the learning objectives), and view evaluations of the course. He can also add to the skill, knowledge, and task hierarchies. He can also access the configuration control capabilities.

The course evaluator reviews the course and inputs his review, evaluation, and suggestions, which only the course authors can view. He is also authorized to examine student models for students taking the course, but without access to their names, so he can see if his hypotheses about the strengths and weaknesses of the course are valid.

The training manager, through the web interface, can edit the skill requirements and other prerequisites for various jobs and add to the skill, knowledge, and task hierarchies. If an ontology conversion is required, perhaps because a job or its vocabulary has changed radically, he can define the mapping from the old hierarchies to the new. He can also input waiver rules and view the resulting course eligibility and projected throughput.

Particular students and/or particular courses or jobs may have designated mentors. A mentor, through the web interface, can view a student's skill mastery model and his career plan. He can answer the student's questions relating to his current job, course, or career.

A student's supervisor, through the web page, can view the student's skill mastery level, submit his own estimates of the student's abilities for both general and specific skills, and update the skills needed and practiced for the student's current job. He will also be notified if the student has not responded to the ITMS in a reasonable period of time.

The permission czar authorizes various users and classes of users to have access to the data and capabilities within the ITMS. In addition to the 8 roles described here, he can create new roles as new combinations of authorized capabilities and data access.

4.2 ITMS Design

The ITMS design is shown below. The heart of the system is the base ontology maintained by the course authors and training managers and which will be referenced by all of the models in ITMS. This base ontology will be described as multiple hierarchies of skills, knowledge and tasks (hereafter simply referenced as "skills"). These are the skills and knowledge required to perform particular jobs. The tasks are the tasks required to be performed in a particular job. These skills may be either general or specific. For example, a particular weapon director may possess the specific skill of recognizing the champagne air-to-air tactic. Another weapon director may possess the more general skill of recognizing any tactic that is attempting to outflank the opposing force. This example also illustrates one type of relationship modeled between elements in the hierarchy – the more-general (or more-specific) relationship. Under this relationship the skill of recognizing any tactic that is attempting to outflank the opposing force will have several children including recognizing the champagne tactic, recognizing the bracket tactic, recognizing the pincer tactic, etc.

The other type of relationship supported between skills is the subtask relationship. The task of a weapon director making picture calls to fighter pilots consists of detecting the enemy tracks, recognizing their tactics, determining which friendly fighters are affected, formulating the proper radio transmissions, then making them. Each of these is a subtask of the making picture calls task.

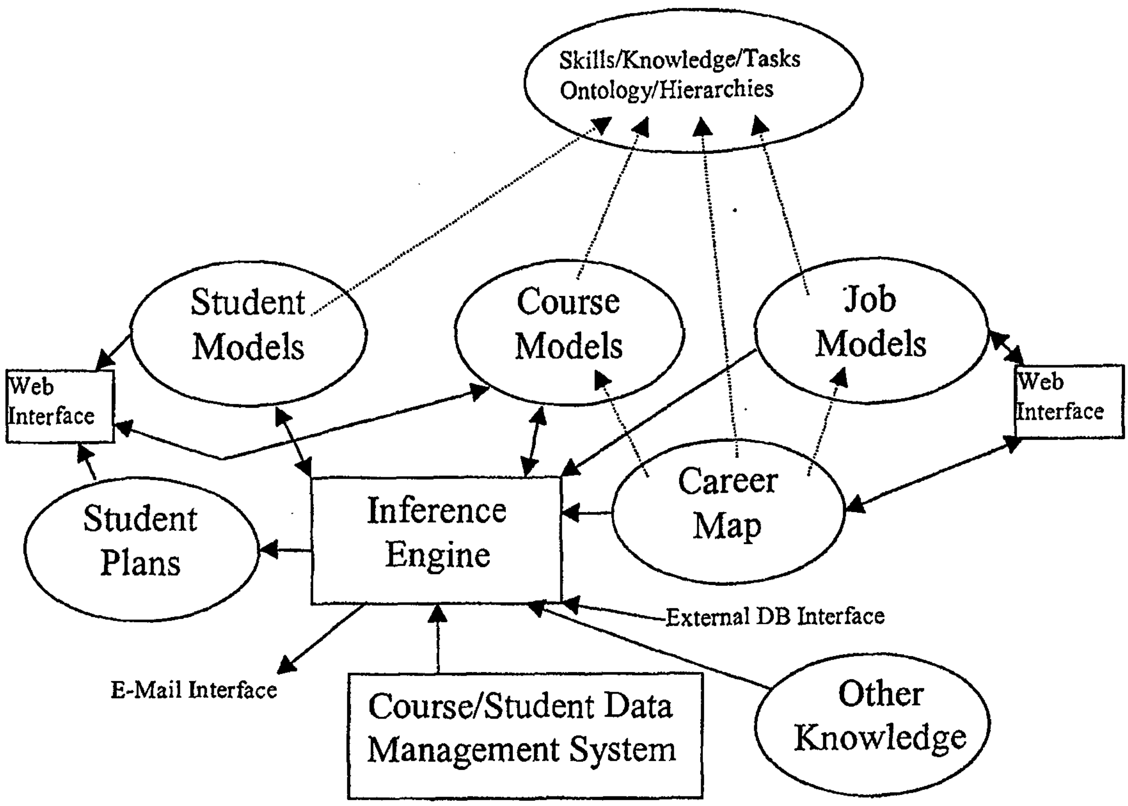


Figure 2. ITMS Design

Course Models

In the figure above, dotted line arrows connote reference links. That is, the course models, student models, job models and career map all reference the skills hierarchies. The course authors maintain the course models and decide at which point the updates are significant enough to warrant that a new version should be defined for the course. The course models include learning objectives which are lists of skills from the skill hierarchy (either general or specific or a mixture of both) as well as the degree of mastery expected from students completing the course. The course model also lists explicit prerequisites which may be any course, job, rank, or other object appearing in the career map. Required prerequisite skills needed to successfully take the course can also be described in the course model using items from the skill hierarchies and degree of mastery required. The course model, in addition to the course author's estimates, will also include ITMS's estimates of the course's ability to make student's achieve various levels of mastery. These are statistical estimates based on constraint satisfaction applied by the inferencing engine. The course model will include estimates from course evaluators as well. The course model (for editing and examination of ITMS's quality estimates) is available to course authors through their web interface.

Job Models

The job models are maintained by training managers and, indirectly, by supervisors. Each job consists of a list of skills from the skill hierarchy (either general or specific), as well as the degree of mastery required to perform the job. If skills are expected to improve during the course of the job or other on-the-job-training is expected to occur, the mastery level expected at the end of the job assignment is also described. These initial estimates are also updated by ITMS as it gathers more data, primarily from supervisors of the students holding the relevant jobs.

Career Map

The career map primarily shows the relationship between the various courses, jobs, and ranks in the domain. Arrows between these objects represent explicit prerequisite relationships. For example, a particular rank may be required to take a particular course, which is required before assignment to a particular job. In the career map, prerequisite arrows would be shown from the rank to the course to the job. Any object may have any number of explicit prerequisites and may be the explicit prerequisite to any number of other objects. Objects can also be implicit prerequisites for each other, by the definition of prerequisite skills described above. The course map objects also include heuristic knowledge as to how fast they can be expected to be accomplished. The career map can be accessed by students who are in the process of determining their career goals and is maintained by a manager for that specific domain. For example, the career map for AWACS weapons directors (WDs) would be maintained by the manager responsible for defining the requirements and prerequisites for AWACS weapon directors and senior director jobs.

Student Models

The student models are generated by the ITMS and basically copy the structure of the skills hierarchy. For example, the student model for a particular AWACS WD would contain the entire hierarchy for the AWACS WD domain, both the low-level, specific skills and the high-level, more general skills. For each skill in the hierarchy, the ITMS will have estimated the mastery of the particular WD in that low or high level skill. The first estimates for a skill are based on the first course or job that develops some mastery in that skill. This estimate is refined with additional data as it becomes available to the ITMS over time. Furthermore, the ITMS will make additional inferences as appropriate. For example, if all of the subtasks for a task are mastered, then it is likely the task itself has been mastered (subject to the ability and need to perform them concurrently). Similarly, if a course teaches the general ability to recognize tactics and, additionally, the student has demonstrated some proficiency in recognizing specific tactics, it can be inferred that he can recognize a variety of tactics (or easily learn to), even if he has not been tested on them before.

Course/Student Data Management System

The ITMS will be able to track and manage thousands or even millions of students. Thus, the ITMS will store the information associated with students and the student model in a database management system (DBMS). Similarly, there will be a lot of information associated with the results of particular students taking particular courses and these will also be stored in a DBMS. This is indicated in the design by the "Course/Student Data Management System" box.

The ITMS needs to get the results of each student taking each course. In the case of resident courses, this would likely occur using the interface to external databases, so that a batch of students who have just completed a course, and whose results are stored in a database, could have their results input electronically, all at once. But in the case of electronic courseware, it would be best to have the courseware automatically send the results to ITMS. SHAI will provide a library of code which course authors can use and easily add to their courseware so that the results are sent back to ITMS when the student completes the course.

Similarly ITMS expects to get results in the vocabulary of the skill hierarchy. The courseware may have the data in a different form, such as listing of correct and incorrect student actions. SHAI has developed existing code which can be customized by course authors to convert student action lists to estimates of the degree of mastery of skills. This code will be packaged and provided to the course authors as well.

Student Plans

The student plans are generated by the ITMS in consultation with the student. The student examines the career map and selects career goals from the objects in it. ITMS then examines his current accomplishments (in terms of mastered skills, courses taken, and jobs and ranks held) and computes what is required, in what order, in how long it is likely to take, and reports this information back to the student.

Other Knowledge

In addition to the models and career maps, there is a large amount of other significant knowledge which can be edited and input into ITMS by users. This includes heuristic a priori decay factors; contact methods with heuristic estimates as to their success probability, likely delivery times and level of effort; proactivity knowledge and training regulations and requirements knowledge, as described below.

One specific type of knowledge represented with ITMS is general knowledge of training requirements for individuals and teams. ITMS includes an intelligent training requirements module which uses this knowledge to determine the training requirements for each individual and team. We use an object-oriented approach to represent training regulations and requirements. Within the ITMS, objects exist which correspond to different positions and teams within the Air Force and which describe the applicable training requirements. Through a multiple inheritance scheme, objects which correspond to individual trainees are dynamically instantiated as members of the applicable classes. For example, an AWACS pilot would be instantiated as a member of the following classes: Air Force personnel, air crew member, pilot, and AWACS pilot. Furthermore, the pilot's team would be instantiated as an air crew and AWACS team. Associated with each class are the data and methods to calculate the training requirements, based on the individual's and team's histories. These requirements are themselves objects that describe the resources needed for the training requirement, or event. The training event might simply be 20 hours of classroom training on applicable threats. In that case, the resources might only be a classroom, instructor, and presentation materials and devices. A more complex training event might be a Defensive Counter Air (DCA) AWACS sortie. The required resources would be an AWACS aircraft, 18 trainees to man the positions, perhaps 10 instructors, a tanker and crew, 4 F-16s to be controlled (along with their pilots), several aircraft and pilots to fly the threat profiles, and enough airspace to play the engagement.

The requirement objects for each individual and team are displayed in the Identified Requirements Editor. These can be accepted or supplemented by the training manager. These are sent to the intelligent scheduler, possibly with the addition of more constraints. ITMS includes an intelligent scheduling capability that manages all the resources under the corresponding training manager's control. Different types of training managers have different types of resources under their control and so the different intelligent schedulers perform different functions, but all will use the same underlying technology.

Intelligent Scheduler

The following example illustrates the complex negotiations automatically managed by ITMS. A training manager in charge of the 25 AWACS weapons director (WD) trainees, is primarily tasked with determining the training requirements for each and managing their schedules. The intelligent scheduler on her local PC is only free to select times for the trainees (subject to their entered availabilities and constraints) but not to schedule/allocate resources under someone else's control. If it was determined that several WDs required a training sortie, a request with their available dates would be sent to the intelligent scheduler residing within the ITMS on the AWACS sortie manager's desk. This scheduler, while having no control of trainees, would have control of the resources controlled by the sortie manager, including AWACS aircraft and maintenance crews. Unfortunately, it does not have control of the required fighters or their pilots and so would have to send requests for those assets to applicable fighter units. If those units were also running ITMS, their intelligent schedulers could automatically respond to the requests. If not, E-mail requests would be sent to the managers of the fighter sorties requesting specific dates and times and/or what existing fighter sorties could be piggy-backed with. The E-mail would include an automated form that facilitates the human responses and is easily machine readable. Eventually, the sortie manager's intelligent scheduler would determine what it thought was the best sortie schedule and send it back to the original requester (in charge of the 25 WDs) for confirmation or continued negotiation. This form of negotiation and interaction is what SHAI has already provided in our existing intelligent scheduling systems.

One of the most important functionalities that the intelligent scheduler (and collaborator) can provide is rescheduling in response to dynamic changes. If an instructor canceled at the last minute, the training manager could request ITMS to find an alternate which would cause the fewest perturbations in the training plan. ITMS can do this since it has access to everyone's schedule and knows each person's whereabouts. In the more extreme case, if a fighter unit cancels, ITMS can automatically determine alternatives and contact them. The manager can interact with ITMS to select the best alternatives.

At any time in the scheduling process, humans can intervene and edit the current schedule or define additional constraints or resources. Inside the schedule editor, an explanation as to why the particular resources and time windows were selected is given. Alternative acceptable times can also be displayed. If the user alters the schedule, the intelligent scheduler will check for violations of constraints or over-commitment of resources (including trainees). Once the schedule is finalized and approved, it is automatically published and disseminated to all applicable parties. Since the intelligent scheduler knows all of the resources required for each event, it is a simple matter to send notifications to the manager of each resource. These can be formatted plots showing graphically, for each resource under the manager's control, when each is committed and for which training events.

Inference Engine

The inference engine infers new information and knowledge and makes decisions. In the diagram above, an arrow directed toward the inference engine implies that it uses that information to make an inference and an arrow directed away from the inference engine indicates that it derives and outputs the indicated information. Arrows in both directions indicate both an input and an output relationship. The inferences take several forms. The mastery of skills by each student must be inferred by the ITMS and placed in the student model. This inference for skills for which data directly exists is more straight-forward. The data can come from many sources including course results, course-independent tests, supervisor evaluations, and follow-on course results. Furthermore, since there are relationships between skills in the skill hierarchy, mastery can be inferred for other skills based on mastery estimates for adjacent ones. For example, the ITMS can infer mastery of all subtasks if the supertask is mastered (and vice versa as in an earlier example). Similarly if a more general skill is mastered, all of the more specific skills underneath it can be considered mastered. Additionally, if the student has shown that he can quickly master a number of sibling skills under a more general skill, it may be safe to assume mastery of the more general skill as well.

These previously described types of inferences are one form based on graphical information. Another is based on the career map. The inferencing engine can use the graphical prerequisite links in the career map to assemble the chronological order of events that the student must accomplish to achieve his goals, given his current accomplishments. The inference engine also examines the skill prerequisites of the goals and subgoals, compares them to the student's current levels or the values expected after taking one of the courses, and determines if additional courses are required to address any skill deficiency.

To evaluate a course's ability to meet its learning objectives, the engine uses a combination of constraint satisfaction and statistical inference and uses data from several sources. Consider a very simple example where are there 4 students and 3 courses. Each course has several learning objectives, several skills, it is trying to teach. After taking the course, each student will have several opportunities to have his relevant skills evaluated. This situation is depicted below. Student A has taken Course 1 and 2 as denoted by the arrows. Course 1 happened to have developed a particular skill, Si. At the end of the course, student A will be evaluated in reference to skill Si and this can be used as input to the process of determining the course's ability to teach Si. This is shown by the "SA, C1, Si Results" box. This box is attached to an arrow that is an extension of the arrow from student A to course 1, indicating that evaluation of skills taught to Student A by course 1 will continue to be evaluated over time. The second box up the evaluation arrow indicates data on student A's skill Si mastery from a job performance review by his supervisor. Finally, the last box shows the results of student A's performance at the beginning of a new course, C11, that requires Si as a prerequisite skill. The Decay box indicates that before taking the course, a six-month delay occurs during which time skill Si decayed some amount. Keep in mind that this structure would be replicated for every skill taught by course 1 to student A. And of course this structure is replicated for every student taking each course.

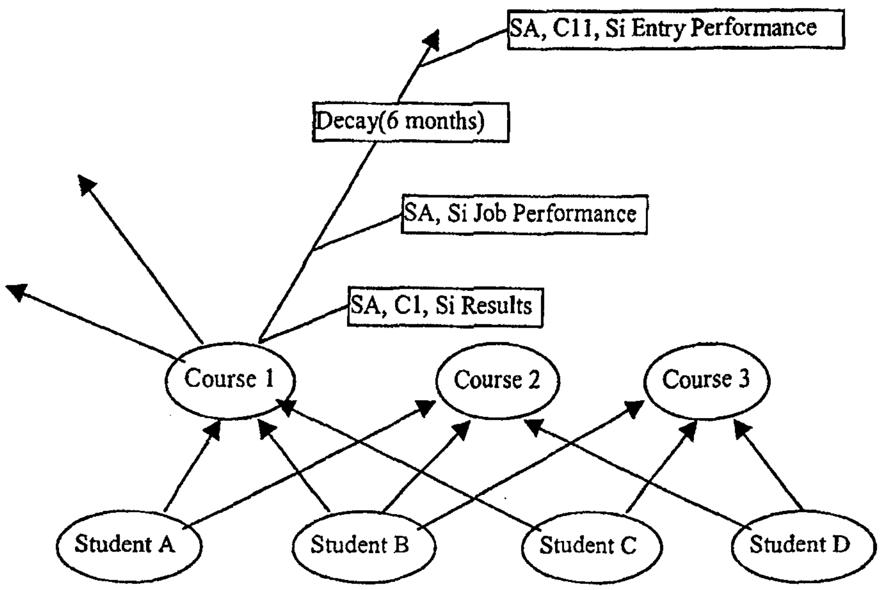


Figure 3. Constraint Satisfaction Network

Constraint satisfaction is used in the following way. The assumption exists that for each skill, a course has some quality in its ability to teach that skill (either specific or general skills). This quality will vary for different types of students. Furthermore, each student has his own unique ability to learn new skills. This can be estimated for a particular student by seeing how well he learns new skills in all of the courses (and other learning opportunities) he experiences. His mastery of skills is given, in a noisy way, by the various evaluations performed on him. Furthermore, if significant time elapses without practicing a skill between when it was learned and when it was evaluated, decay will be assumed to occur. There will be an apriori estimate of this decay for each skill, but it is assumed to vary somewhat for different students. The problem becomes how to most consistently label the graph to explain the various (noisy) evaluation results. This is both a statistical and a constraint satisfaction problem. Any course or student cannot be considered in isolation, as the figure above shows. For example, if the results for course 1 are poor, it may be caused by a poor course, poor students, or a mismatch between the specifics of the course and the attributes of the students. If the students have done well in other courses then the second hypothesis is eliminated. Unless the students are very similar to each other, the third hypothesis cannot be the sole explanation either. Thus, to determine which of the three (or which combination of the three) is most appropriate, the data for all students and all courses must be considered simultaneously. Constraint satisfaction techniques were developed for precisely this type of problem.

The inference engine also reasons about time. When it sends an E-mail notification to which it expects a response, it schedules an event in the future to "timeout" if it has not received the response and take the appropriate action. If the response occurs before that scheduled event, the event is cancelled. Similarly, it will use the decay heuristics to determine when a student's skills should be checked, if he is not exercising them. When that day arrives, the student's recent history is examined to determine if in fact, skill decay has likely occurred, and if so, to take the appropriate action. This is how the inference engine achieves proactivity.

As mentioned previously, the inference engine may decide to make proactive notifications using an E-mail system, so an interface to such a system is shown in the design. Additionally, data from an external database system must occasionally be processed and thus, an interface to these is provided as well.

4.3 Functionality

The ITMS is a general capability that can be customized by the users to manage any system of training courses. This occurs by first creating the skill hierarchy then models for the courses and jobs. The process is completed by input of the career map and miscellaneous heuristic knowledge.

Student-Related Functionality

One of the ITMS's most important functionalities is the pro-active notification of students. This typically occurs via E-mail with an acknowledgement expected. ITMS will follow up with additional E-mails and/or regular mail, if required. ITMS will eventually notify the student's supervisor, if it receives no response. If the student happens to log on his web page during this period, ITMS will explicitly request updated E-mail and contact information. Student response can be via E-mail or the Web. Based on the results of user requirements analysis, we may also provide ITMS the ability to make notifications by telephone and receive some types of responses by phone.

The notifications will include telling them when they need to take prerequisites for future courses and telling them that they're falling behind in completing the prerequisites or other career goals. ITMS will proactively notify them that certain skills may have degraded and need to be evaluated and possibly refreshed, that their next assignment requires a different skill set and therefore refresher or additional training or scenarios, and that courses or job requirements have changed and additional training is needed to stay up-to-date.

ITMS will also provide information to the students via their personal ITMS web page including their strengths, weaknesses, and progress. This will be a bar chart that shows their mastery level of relevant skills. These bar charts will be hierarchical and follow the skills hierarchy. Thus, the first bar chart will correspond to the first level, or breakdown, of the student's skills. The student can then click on any particular bar to see that skill's subskills expanded (based on either the subtask or the more-specific relationship) in its own bar chart. Any of those skills can be similarly selected and so on. The student's web page will have a "What's new" section and a "What's new for him" as well which would contain information about new courses, new versions, or new knowledge required for his specific job or based on courses he has taken.

The ITMS web page will also include a career counseling section where the student can view the career map and select career goals and timelines for achieving them. The ITMS will provide advice as to what timelines are reasonable and achievable. The ITMS will determine, from their career goals and already achieved skills, ranks, jobs, and courses, and from the career map, what subgoals are required to meet the student's objectives and in what order. This will be based on the explicit prerequisite relationships as well as required prerequisite skills. If any of the student's skills meet the required prerequisite levels, the ITMS will find appropriate courses that can build the skill level from what the student possesses to that required for some objective.

The ITMS will proactively question the students (and the ITMS will expect answers via E-mail or web page). It will get feedback on each course they've taken as to its ability to build mastery in the specific and general skills as well as the prerequisite skill levels required. It will get feedback on their current job, what it entails and their ability to meet it. It will give tests and evaluations, if the system suspects skill decay, and provide remedial refresher courses, if appropriate. ITMS starts with heuristic apriori decay constants for each skill but learns actual constants based on skill, skill type, and individual soldier.

Explanation Capability

The inference engine, which makes all decisions in ITMS, will record rationale for each of its decisions. These then provide the basis of an explanation facility for students and other users. For example, the student could ask why the ITMS has included a particular course in his career plan and it might respond with the explanation that it was a prerequisite for one of the prerequisites for one of his career goals. He might ask why the ITMS believes certain skills have decayed and the ITMS would respond with a description of what it believes the student's job currently is and what skills are not practiced by that job along with the rate at which it believes those skills decay for that student. A supervisor might ask for an explanation for why the ITMS as estimated the mastery of a certain skill for the student to be a particular value. The ITMS would respond with a description of how it was calculated and where the supporting data come from. Similarly a course author might ask for an explanation for the ITMS's calculation showing the degree to which the course was meeting one of its learning objectives.

Supervisor and Mentor-Related Functionality

The ITMS will proactively question the supervisors (who answer via E-mail or their web page). It will get feedback on soldier and the preparation for his current job provided by courses he has taken. It will get updated job requirements, both general and specific, for the particular position. It will provide the same bar chart type functionality to view the skills of students under their supervision, subject to the authorization of the training manager. Similar functionality will be provided to the student's mentors. These will also have facilities for receiving and answering student questions regarding their job, career, or courses that they are taking.

Course Author-Related Functionality

Course authors will be provided a graphical editor to maintain the skill hierarchies. The editor will show graphically either one or both hierarchical relationships simultaneously, and provide user-friendly point and click methods for adding new skills, and linking them to other skills through one of the two relationships – subtask or more-specific. A similar interface will exist for selecting which skills from these hierarchies are being taught and to what level by courses under the author's control. In addition to the skills taught, the author will also specify the prerequisite skills and degree of mastery required for successful entry into the course. These skills may be specific and concrete or more general and abstract in nature. He will also graphically specify prerequisite relationships between his course and other courses, jobs, or ranks (by editing the career map). Additional course information includes the version, possible scenarios and their attributes, required hardware or software or other constraints.

He will also be able to get evaluations of his course in bar chart format where his estimates of the prerequisite and resulting skills of the course are compared to evaluator's assessments and actual results from students who have taken the course. Their improvement (or lack thereof) of skills, after having taken the course, will be based on supervisor evaluations of students job performance and the course's ability to prepare them, student self and course evaluations, and performance in subsequent courses. The evaluations will consider the student's improvement in both specific and general skills. Data-mining techniques will also be used to try to differentiate course value between different types of students. In these cases, the ITMS will make suggestions for improvement by identifying course areas that are weak (which skills are not being taught as well as expected) and which course areas are weak for different types of students. It will also suggest when additional scenarios might be needed to cover aspects of a course that aren't currently covered or not covered by enough scenarios based on student use and failure patterns. The ITMS will also provide the author with how many students have taken which versions and/or scenarios and which did the best later.

The ITMS will also provide, to a group of course authors, course configuration management capabilities. Configuration management refers to aiding the courseware development process by tracking the different versions of the separate files that make up the courseware, making sure that all the development team is using the most up-to-date versions, and that the final released product has these most up-to-date versions of files. ITMS will maintain a directory of "published" courseware files which represent the currently accepted version of a course. Authors then "checkout" these files to make updates and the ITMS keeps track of who has what file and when it was checked out. The file then becomes read-only for other team members and they are warned, when they view it, that it is currently being revised, when the revision was started, and who is doing the revision. The ITMS would also archive old versions of files. This scheme keeps authors from making parallel changes to the same files, while continuing to let them reference them in their own work. When the revisions are made an approved, the new version of the file is added back to the directory.

Training Manager-Related Functionality

Training managers will be provided a graphical editor (similar to the course authors) to maintain the skill hierarchies and to edit the skill requirements of jobs, as well as the skills developed or practiced by the jobs or expected on-the-job training. They will also have similar graphical capabilities to edit the career map, though their focus will tend to be on the jobs as opposed to the courses. The ITMS will proactively determine if there are skills required for jobs for which no course or other learning event develops the required skills to the required degree of mastery and notify the training manager.

The ITMS will provide the training manager with an overall view of the students, courses and jobs in his specialty. It will provide him graphically with how many students are currently at each point in the career map. It can also project these numbers into the future, based on how filled each spot is and the time required for the average student to achieve various milestones as well as capacity constraints which might limit throughput. The ITMS can also accept waiver rules from the training manager and show how this affects the particular course in terms of eligible students and expected graduations (based on percentages who expect to pass, given the waivers) or how it affects the overall system, into the future. The ITMS will also provide how many students have taken each course, version, and scenario.

4.4 Innovations

The ITMS includes several innovations:

- ITMS is intelligent. It makes decisions. It "remembers," not just stores, information and knowledge (in the sense that it keeps information in working memory and reacts when certain events do or do not occur). It is proactive. All aspects of the system are stored in an explicit knowledge representation which allows end-users to edit and modify all aspects of the system, subject to the appropriate authorizations, of course.

- ITMS addresses the time span that encompasses an entire career.

- ITMS acknowledges the concept that students, courses, and jobs change over time, so that the corresponding Student Models, Complex Course Models, and Job Models must change as well, even while the history and content of the old versions is preserved.

- The ITMS utilizes user-editable hierarchies of skills, knowledge, and tasks as a basis for the course, job, and student models. Both subtask (to fix a piece of equipment requires the subtasks of trouble-shooting and repairing) and more-specific (the ability to fix a specific radio is a more specific skill compared to the ability to fix any radio) relationships are supported in the multiple inheritance hierarchies.

- ITMS performs several kinds of inferencing. It makes inferencing based on graphical descriptions (prerequisite links and hierarchical relationships), using Constraint Satisfaction, and based on statistics.

- ITMS makes proactive decisions and actions, including proactive E-mail notification.

- Training requirements, resources, events, trainees, instructors, and other ITMS objects, are not just data, but actual intelligent entities which facilitate many different uses. Just as they schedule their corresponding real world object themselves, they could also be made to provide explanations of their actions for training or optimal resource acquisition purposes.

- No one has previously applied advanced scheduling techniques to the training management problem before.

- The concept of distributed collaboration of a mix of automated and human decision makers in separate organizations and locations is innovative. Although applied to a few problems, it has certainly not been applied to training management systems.

5.0 Existing Training Management Systems

Little or no research has been performed for training management system and no system has employed any Artificial Intelligence techniques. Therefore the related work consists primarily of the many training management software systems that have been developed and marketed. These systems are called by various names including training management systems or software, education management systems, computer-managed instruction (CMI), or training administration systems. There are currently over 60 such systems being marketed [Hall 1998]. Companies marketing products include Allen Communication, Asymetrix, American Training International, CBT Systems, Cytation Corporation, DK Systems, Geometrix, Informania, Integrity Training, ITC Learning Corporation, KnowledgeSoft, Lasso Communications, Inc., Learncom, Inc., Macromedia, NETg, On Tour Multimedia, Oracle, Pathlore, Plateau, Saba Software, Inc., Saratoga Group, Silton-Bookman Systems, Inc., Syscom, Inc., Teamscape, TTG Systems, Inc., Micromedium, and Infotec. The most popular products being marketed include Pathware, Librarian, Manager's Edge, Ingenium, Registrar, TrainingServer, AdminSTAR, SkillVantage Manager, PHOENIX, and World Trak.

These systems do not begin to meet the complex needs discussed here and do not contain intelligent features. These systems are primarily networked database systems and store data relating to course catalogs, class schedules, enrollment, student information, transcripts, class evaluations, homework, self-assessments, course authoring, content management, grades/test scores, and rudimentary skills. The primary benefit they provide is that of a pre-customized DBMS with existing interfaces defined to the vendor's own courseware offerings or authoring tools. The primary disadvantages are that they do not attempt to track higher level skills and they do not exhibit intelligence, decision making, or proactively, leaving these functions to the training managers or the students themselves.

6.0 Future Work

6.1 Phase II

The ultimate goal of the Phase II effort is to aid the training managers. The final system will reduce their work load, improve the utilization of scarce resources, and reduce training management lapses. The primary Phase II objective is to develop a full-scale, operational version of ITMS in Phase II. By working closely with Air Force, other DOD units, and commercial training managers and performing an analysis of the requirements of other commercial potential clients, our implementation effort can be directed most appropriately and therefore most efficiently for commercialization. Since the ITMS will be implemented in three major releases, the Air Force and other users will have the opportunity to use it operationally early in the project and provide us the necessary feedback to perfect it during Phase II. In order to allow operational use, we will need to interface ITMS to several existing database systems as described in Section 3.3, Task Descriptions.

6.2 Potential Applications

The primary Phase II project results will be a full-scale, operational Intelligent Training Management System (ITMS) developed in cooperation with several different users. The ITMS will have immediate use throughout the DOD and Federal government. In fact, it has significant support from current DOD training managers. For example, several officers at Tinker AFB in Oklahoma have expressed a strong desire for an ITMS and discussions with them had a strong influence on the Phase II design. They will be some of the users of the Phase II system, *during Phase II.* The US Army's Distance Learning Center at Fort Huachuca, Arizona, has also expressed much interest in the ITMS and plans to use the Phase II system, *during Phase II.* The individuals charged with managing the training process for the Navy are the ships' executive officers (XOs). Commander Pinto, the XO on the USS Paul Hamilton (DDG-60, an AEGIS Destroyer) has stated that training management is one of his primary problems. He has even begun the process (coincidentally) of requesting that the Navy upgrade its training management software, which is not currently at an acceptable level of capability. Thus during Phase II we will have operational users throughout the DOD to make sure the resulting system is beneficial to the government.

Several other great opportunities to marketing the ITMS to the government exist. These primarily relate to the fact that SHAI is one of the premier Intelligent Tutoring System (ITS) developers, and thus has a large base of customers who are interested in training management. For example, our Tactical Action Officer (TAO) ITS, currently in operational use by the Surface Warfare Officers School (SWOS) and onboard the Paul Hamilton, was recently selected by the Navy for use onboard all AEGIS ships. These six dozen ships all have the same training management problem described by Commander Pinto, and since they will all already be using one SHAI product, it will be straight-forward to introduce ITMS to those same ships.

Similarly Paul Losiewicz, of the Air Force Research Laboratory is involved in Air Force intelligence training. He is extremely interested in both our ITS authoring tool and this ITMS project. Furthermore, one of our committed Phase II users is the US Army's military intelligence training group at Fort Huachuca. They already train many Air Force intelligence specialists and have a good relationship with Goodfellow AFB, the primary intelligence training center for the Air Force. Goodfellow AFB also trains many Army intelligence specialists. With this kind of cross training relationship, the ITMS can be expected to quickly migrate from one center to the other.

SHAI is currently developing an ITS authoring tool for use by NASA training managers to create ITSs to teach astronauts the skills to operate in-space experiments. These training managers have the same management problems that ITMS will address.

There are several potential commercial applications. The commercial corporate training industry is currently $62.5 billion for companies with over 100 employees [Training Magazine 1999]. Even only allocating 3% to the management function leads to $1.9 billion in training management costs. An ITMS can greatly reduce much of these costs, indicating that a substantial market exists. This is further validated by the large number of training management systems currently marketed.

Many large corporations, especially those involved with possibly life threatening activities, also have complex training requirements and would benefit by ITMS. Examples include nuclear power, airlines, toxic waste handling and clean-up, chemical factories and oil refineries. Any organization with complex training requirements would benefit from ITMS. Accordingly, Esteem Software Incorporated, our highly successful commercialization partner for other endeavors, has agreed to market the ITMS to their substantial customer.

SHAI has identified the commercial training industry as its primary marketing target and written our business plan around this assumption. Accordingly we have hired a Director of Business Development, Rick Row, whose resume given below in Section 6.0, to pursue it, full-time. He has begun to establish relationships with many of the vendors of training systems and services including CBT (largest vendor of training system to teach software operation), Wicat (largest commercial vendor of aircraft simulations for pilot and maintenance training), Flight Safety (largest commercial provider of aviation training services), Raytheon Training, and NETg. Furthermore he has already identified some 60 training management systems currently being marketed. Since the ITMS encompasses technology significantly beyond all of them, each is a potential partner for licensing our technology to provide it to their clients, through integration with their products. A few success stories during the Phase II will ease the process of approaching these companies. Furthermore, the next 18 months should see a significant shakeout of these dozens of vendors so that it will be clearer with whom we should license our ITMS technology. The Phase II proposal, explicitly includes the task, "User Requirements Definition," which itself includes steps to investigate commercial requirements and existing tools, partly with an eye toward future partnering arrangements. While at EPRI, Mr. Row arranged intellectual property licenses with manufacturers for EPRI technology resulting in $500,000 annual revenue. We expect him to make similar arrangements for the ITMS with vendors of training and training management software and services.

Meanwhile, as partly described above, SHAI has sold millions in intelligent tutoring system products and services. We will approach these government and commercial customers, all of whom also have training management system requirements. By involving them in the Phase II Knowledge Engineering, User Requirements Definition, Design, Installation and Training, and User Evaluation and Feedback tasks, we will ensure that the final Phase II ITMS is both commercially viable and useful for DOD training managers. Since we will have three ITMS releases in Phase II, there will be ample opportunity to incorporate their feedback. Since we will have operational DOD and commercial users, during Phase II, we will have several success stories which we can use to approach our other ITS clients, training management system vendors, and training system and service providers.

Our commercialization strategy has many facets. From our previous experience, we know that commercialization activities cannot wait for the Phase II project to end. SHAI has developed an SBIR commercialization process that begins with the Phase I presolicitations. We identify which topics have the most commercialization potential for SHAI and then pursue those aggressively. This topic offered great potential because it represents the intersection of two of SHAI's strong areas, which have heretofore been completely separate; intelligent scheduling and training. This project effectively leverages off our successes in these two fields and will allow us to approach our training customers with another product and/or service, as described above.

Concurrent with Phase II, we will perform market research in support of the Phase II task, User Requirements Definition. That task includes defining functionality which will make the ITMS more commercially viable. The market research will provide focus for the set of commercial users who are most likely to buy an ITMS. The requirements of these users can be folded in with those of our currently committed users. Concurrent with Phase II will be the development of features required by the commercial marketplace and development of contacts to sell the resulting ITMS.

There are thousands of organizations which could benefit from ITMS. There also appears to be little competition. Many training management systems exist, but these are not automatic, merely logging and keeping track of a user's decisions, primarily regarding attendance and scheduling. Most of these don't even do deconflicting. Thus, our ITMS will have the significant benefit over competitors of being largely automatic and will also be more tailored to complex training requirements.

There are three different business plans as a result of this project. The first is to sell the ITMS itself. The design will be flexible enough that users will be able to define their own kinds of positions (jobs), teams, trainees, skills and knowledge, tasks, training requirements, training events, resources, etc. Once a user has entered this knowledge, the ITMS will be able to automatically track students' skills, proactively notify him of new courses, prerequisites, or if he's falling behind; determine requirements; schedule training events (including needed resources); and track results for individuals or teams. Industries with complex training can be approached directly with ITMS, or we could sell it through companies which currently provide training to them (such as Wicat and Flight Safety, who both serve the aviation industry). These would also leverage off our existing extensive Intelligent Tutoring system marketing efforts.

SHAI has completed intelligent scheduling system projects for NASA and is starting others. SHAI scheduling products are already in use by several organizations and we are expanding our market for such tools. We anticipate that this effort will result in additional scheduling algorithms that we will be able to incorporate into our existing scheduling products, thus increasing the benefits they provide and their value.

Finally, ITMS can be used as a basis to create customized ITMS solutions for individual commercial organizations. Because it is designed for low cost application to new training management problems, we can customize it at a low cost.

7.0 References

[Agrawal et al., 1993] "Mining association rules between sets of items in large databases, "In Proc. of the ACM SIGMOD Conference on Management of Data, 1993.

[Allen, 1984] Allen, J., F., "Toward a General Theory of Action and Time", Artificial Intelligence, 23, 1984.

[Beck et. al., 1997] Beck, J., Davenport, A., Sitarski, E., Fox, M., "Beyond Contention: Extending Texture-Based Scheduling Heuristics," AAAI-98 Proceedings, AAAI Press, Menlo Park, CA, 1997.

[Charniak, 1991] "Bayesian Networks without Tears," AI Magazine, 1991.

[Cheeseman et al., 1988] "Autoclass: A Bayesian Classification System," 5th International Conference on Machine Learning, pp. 54, 1988.

[Fox, 1983] Fox, M. S., Constraint-Directed Search: A Case Study of Job Shop Scheduling, Ph.D. thesis, Carnegie-Mellon University, 1983.

[Hall, 1998] Hall, B. "Training Management Systems: How to Choose a Program Your Company Can Live With," Brandon Hall Resources, http://www.brandon-hall.com, October 1998.

[Henke, 1996] Henke, A., Artificial Intelligence Techniques for Mission Planning and Scheduling Automation- Final Report dated July 1996, Stottler Henke Associates, Inc.

[Henke & Stottler, 1991] Henke, A.L. and R.H. Stottler. (1991), Automated Manifest Planning: An Artificial Intelligence Approach; Phase I Final Report. Research Report. Stottler Henke Associates, Inc., Belmont, CA, (July).

[Henke & Stottler, 1992] Henke, A.L. and R.H. Stottler. (1992), "Artificial Intelligence Techniques for Scheduling Space Shuttle Processing", Proceedings of the Florida AI Researchers Symposium, Ft. Lauderdale.

[Henke & Stottler, 1995] Henke, A., Stottler, R., Artificial Intelligence Techniques for Flight Activities Planning, Stottler Henke Associates, Inc., 1995.

[Knorr & Ng, 1996] "Finding Aggregate Proximity Relationships and Commonalities in Spatial Data Mining," IEEE Trans. On Knowledge and Data Engineering, 1996.

[Laird and Saul, 1993] "Sequence Extrapolation," Machine Learning Conference Proceedings, pp. 1120-1125, 1993.

[Maher, et. al., 1997] Maher, T., Green, J., Goan, T., Artificial Intelligence Techniques for Flight Activities Planning - Phase II Final Report, May 1997, Stottler Henke Associates, Inc.

[Mannila, 1995] "Discovering Frequent Episodes in Sequence," Knowledge Discovery in Databases Proceedings, 1995.

[McVey et. al., 1997] McVey, C., Atkins, E., Durfee, E., Shin, K., "Development of Iterative Scheduler to Planner Feedback", AAAI-98 Proceedings, AAAI Press, Menlo Park, CA, 1997.

[Morgado & Martins] Morgado, E., Martins, J., "CREW_NS: Scheduling Train Crew in The Netherlands," AAAI-98 Proceedings, AAAI Press, Menlo Park, CA, 1997.

[Shatkay & Zdonik, 1996] "Approximate Queries and Representations for Large Data Sequences," ICDE 1996.

[Training Magazine, 1999] "Industry Report 1999", Oct. 1999.

[Zweben & Eskey, 1989] Zweben, M., Eskey, M., "Constraint Satisfaction with Delayed Evaluation" Eleventh International Joint Conference on Artificial Intelligence, Detroit, Michigan, August, 1986, pg. 875.

[Zweben & Fox, 1994] Zweben, M., Fox, M. S., eds., Intelligent Scheduling, Morgan Kaufmann, 1994.

Appendix A Phase I Prototype Design

Introduction

This document is a design specification for the Phase I part of the ITMS project. The purpose of this part of the project is to implement and demonstrate a proof-of-concept system that will utilize AI techniques to improve the training management process (i.e. increase training efficiency, streamline the course revision process, etc.) This project will be done in Allegro Common Lisp dynamic object oriented system.

A.1 ITMS architecture

The intelligent training management system was originally conceived as a standalone application, to be developed in Lisp under Windows. Further knowledge elicitation revealed that there was a defined need for a Web interface to much of the ITMS functionality, and so the Web interface (implemented as a number of CGI scripts written in Perl) was subsequently added to the existing Lisp application. The current system is schematically shown in figure 1. The Phase I setup is less than ideal for a number of reasons, and a number of enhancements will be made in Phase II that should increase the usability of the current ITMS prototype.

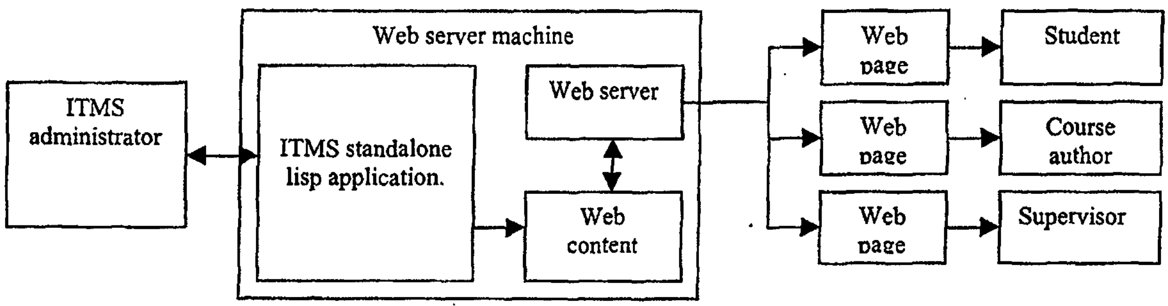


Figure 1.

1.) Single language. Since ITMS consists of two distinct pieces, there is a certain overhead involved in converting data from the form the lisp application understands to the form the web interface understands. This overhead will be eliminated entirely in Phase II, since ITMS will consist exclusively of a Web interface over a database and a reasoning engine.

2.) Perl and Apache integration. In Phase I, the Apache web server was used to generate web content generated by CGI scripts written in Perl. The basic setup used in Phase I was such that the entire perl interpreter, the cgi script, and all the data files used by the script had to be loaded into memory each time the user submitted an HTTP GET or POST request (i.e. each time the user pressed a button on the Web interface). This presented significant overhead, which can largely be eliminated by the use of mod_perl, a perl module that provides Perl/Apache integration. After mod_perl is installed and compiled into Apache, the perl interpreter itself is always resident in memory and doesn't need to be loaded each time, and cgi scripts and data files are cached in memory as well after their first execution. Installation of mod_perl is known to result in two-fold increase in response times.

3.) Script separation and modular design. Additional speedup can be obtained by separating script functionality into distinct scripts (so only a portion of the scripts need to be executed during each GET or POST request).

A.2 ITMS skill graph structure

(not implemented yet)

Predicates (links) relating graph nodes:

Subtask(A, B) True if A is a subtask of B

Example: Subtask(Disassemble power supply, Repair power supply)

PrerequisiteOf(A, B) True if A is a prerequisite of B

Example: PrerequisiteOf(File systems, Networked file systems)

ParentOf(A, B) True if A is a parent of B

Example: ParentOf(Analytic ability, Computer systems)

SkillLevel(A, X) True if the skill level of A is X

Example: SkillLevel(MechanicalRepair, 50)

Inferences we want to make:

SkillLevel(B, L) $\wedge$ PrerequisiteOf(A, B) $\rightarrow$ SkillLevel(A, L)
$\forall$X (Subtask(X, A) $\wedge$ SkillLevel(X, L)) $\rightarrow$ SkillLevel(A, L)
Subtask(X, A) $\wedge$ SkillLevel(A, L) $\rightarrow$ SkillLevel(X, L)
ParentOf(A, B) $\wedge$ SkillLevel(A, L) $\rightarrow$ SkillLevel(B, C * L), $0 < C < 1$.

A.3 ITMS GUI

ITMS GUI will be a conventional Microsoft Windows menu-based GUI. The current GUI structure is shown in figure 2.

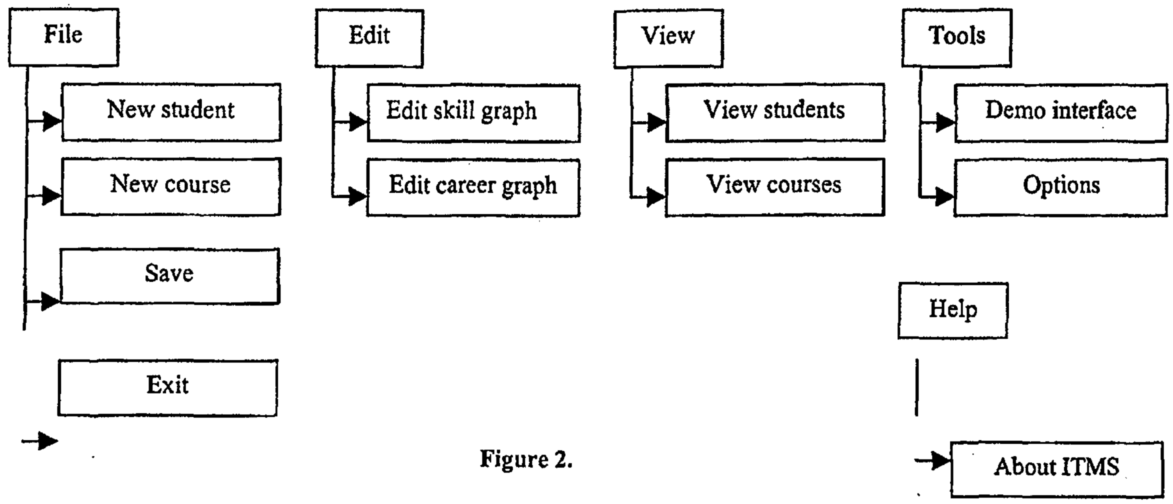


Figure 2.

Figure 3 shows the dialog boxes that make up the ITMS GUI, and how the user navigates through them.

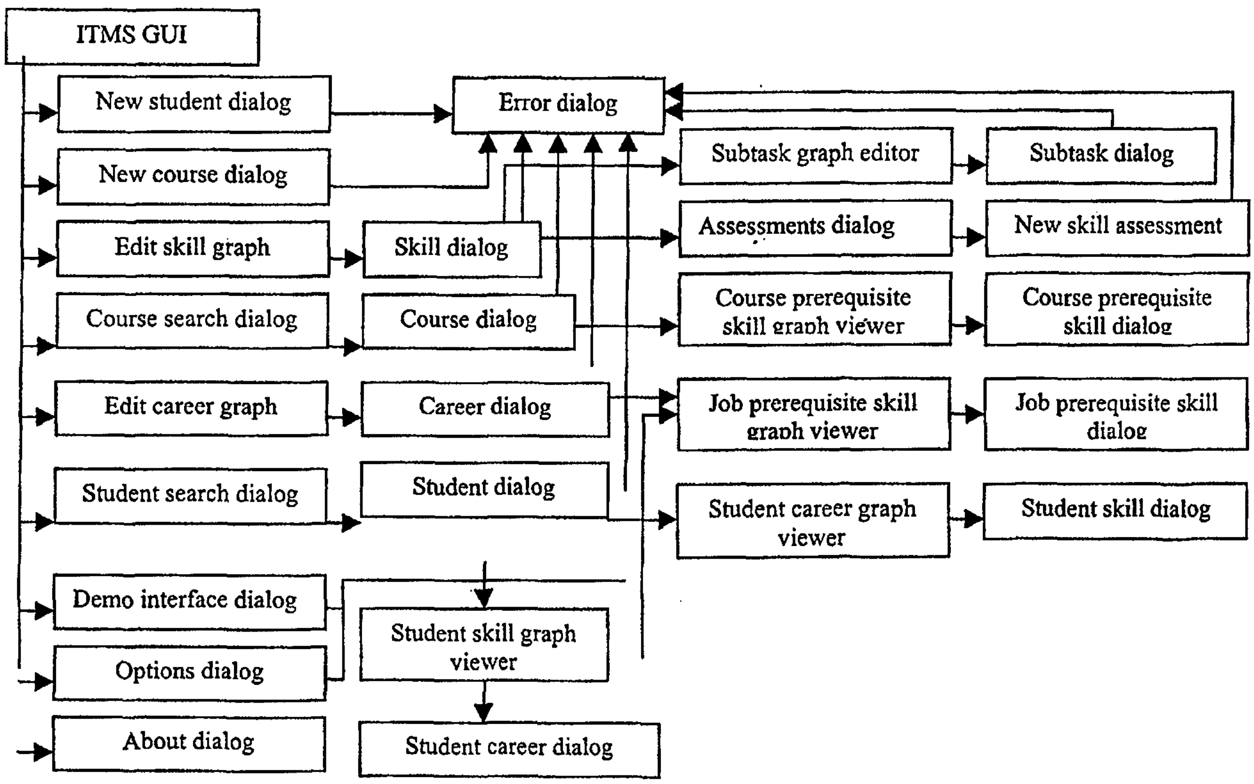


Figure 3.

A.4 ITMS classes and data structures (lisp application)

Class ITMS

;This is the ITMS inference engine.

Slots:

```
:Career-graph        ;the global copy of the career hierarchy (jobs, titles, rank, etc.)
:Skill-graph         ;the global copy of the skill hierarchy

;; These slots can end up consuming large amounts of memory, and so will end up being obtained ;;
from disk as needed (in Phase II at any rate).

:Students            ;student models currently in the system.
:Courses             ;course models currently in the system.
:Schedule            ;a list of events sorted by date.
:Mailbox             ;e-mail interface.
:DL-supervisor       ;contact information for the Distance Learning supervisor.
:Old-versions        ;a list of older versions of career milestones.
:Time                ;current time
```

Methods:

```
;; General methods.
(read-itms      file)             ;reads the ITMS structure from a datafile.
(write-itms     itms    file)     ;writes a given ITMS structure to a datafile.
(init-object    object  stream)   ;initializes ITMS from a given input stream.
(print-object   object  stream)   ;writes ITMS to a given output stream.

(update-skills  graph)            ;propagate global skill hierarchy changes
(update-career  graph)            ;propagate global career graph changes

;; Event methods

(clear-event     id)     ;clears the event associated with id from the event queue.
(trigger-events  date)   ;triggers events scheduled for a particular date.
(schedule        event)  ;Schedules an event for some future date.

;; Scheduling methods    ;These methods schedule various events to occur at a specific time in
                         ;the future.

(schedule-career-survey   student  career-name   days)
(schedule-falling-behind-message   student days  plan-days)
(schedule-timeout-message          address cause days)
(schedule-skill-decay     student  days)

;; E-mail creation methods ;These methods create e-mail message that ITMS sends out.

(make-new-course-version-message)
(make-new-career-version-message)
(make-falling-behind-message)
(make-timeout-message)
(make-describe-plan-message)
(make-no-legacy-career-string)
```

(make-create-more-versions-message)
(make-legacy-career-string)
(make-skill-and-course-string)
(make-not-enough-time-message)
(make-goal-accomplished-message)
(make-useful-courses-message)
(make-create-useful-courses-message)
(make-ready-message)

;; E-mail sending methods ;These methods send e-mail messages out.

(send-useful-courses)
(send-ready-message)
(send-course-info-message)
(send-student-info-message)
(send-itms-help-message)

;; E-mail response methods ; These methods are called by mailbox class in response to
; certain e-mails ITMS receives.

(update-skills-and-decay)
(update-career-goal)
(career-completed)
(update-job-skills)
(update-course-skills)
(update-course-enrollment)
(career-assignment)

Class ITMS-gui

;Class defining the main GUI for ITMS.

Slots:

:itms ;the slot holding the itms engine itself.
:new-student-dialog ;Various dialog box classes that ITMS displays in response to user ;action.
:new-course-dialog
:skill-graph-editor
:career-graph-editor
:student-search-dialog
:course-search-dialog
:demo-interface-dialog
:option-dialog
:error-dialog
:about-dialog

Methods:

(initialize-instance :after) ;This method sets a number of variables ITMS-gui depends on.

(new-student) ;These methods are invoked whenever the user selects the appropriate
;action from the menu.

(new-course)
(edit-skill-graph)
(edit-career-graph)
(view-student)

(view-course)
(demo-interface)
(options)
(about-itms)

Class Student

;Class defining the student in ITMS.

Slots:

:first-name	;Personal information...
:last-name	
:e-mail	;The student's e-mail address (very important).
:address	
:city	
:state	
:zip-code	
:supervisor	;The student's supervisor contact information.
:old-skill-levels	;Student's old skill levels (used for skill decay estimation)
:current-list	;The list of current career assignments.
:goals	;The list of career goals.
:plan	;A graph representing the student's plan.
:courses	;A list of courses taken by the student.
:career	;A list of career milestones taken by the student.
:skills	;The student's local copy of the skill graph
:os	;Student's OS
:cpu	;Student's CPU class
:memory	;How much memory the student has.
:disk	;How much disk space the student has.
:cd-rom	;Does the student have a CD-ROM?
:internet	;Does the student have internet access?

Methods:

(update-skills)	;Update student skills to reflect global hierarchy changes
(clear-skills)	;Remove skills no longer present in the global hierarchy
(update-career)	;Update student career to reflect global hierarchy changes
(can-run job)	;Returns true if the student's hardware/software supports a given career ;milestone.
(ready job)	;Returns true is the student has sufficient skills to undertake a given career ;milestone.
(decay-skills)	;Decays the student's skills.

Class Job

;Class representing a career milestone (a course, a job, or a rank, in our domain).

Slots:

:name	;The job name
:number	;Version number
:type	;job, course, or rank.
:supervisor	;Contact info for the supervisor.
:length	;Minimum length one must spend on the job.

:students	;The students previously enrolled here.
:skills	;Skills required and taught by this milestone.
:skills-estimated	;Skill estimates for this milestone.
:os	;OS needed for this milestone (if any).
:cpu	;CPU needed for this milestone (if any).
:memory	;memory needed for this milestone (if any).
:disk	;disk needed for this milestone (if any).
:cd-rom	;Does this milestone require a CD-ROM.
:internet	;Does this milestone require internet access.

Methods:

(update-skills)	;updates skills in response to changes in the global hierarchy.
(clear-skills)	;clears skills no longer present in the career hierarchy.

Class Message

;Class representing an e-mail message.

Slots:

:data	;The message itself.
:attachments	;Any attachments to the main message body.

Methods:

(Send message)	;Sends a message through the e-mail system.

Class Skill

Slots:

:name	;The skill name
:decay	;The default decay constant for the skill
:expertise	;the expertise of a particular student in a skill.
:level-needed	;skill level prerequisite for a particular job or rank.
:level-developed	;skill level developed at a particular job.
:assessments	;a list of ways to assess the skill (sorted by assessment cost).

Methods:

Class Graph

;A non-intrusive implementation of a directed graph container. Supports arbitrary data structures ;as nodes, can apply functions to nodes in hashed, or depth first order (so far). Adds and removes ;nodes efficiently (with a function that cleans up edges that is called after a series of node inserts ;and removals).

Slots:

:roots	;the ids of root nodes of the graph
:acyclic-p	;is true if the graph has no directed cycles
:nodes	;the nodes of the graph
:depth	;the maximum depth of the graph

Methods:

```
(insert-node node id parent-ids child-ids)
(remove-node id)            ;removes a node and all incoming and outgoing edges.
(remove-subtree id)         ;removes a node and its entire subtree.
(remove-node-splice id)     ;removes a node, but connects all of its parents to all of its children.
(insert-edge parent-id child-id)
(remove-edge parent-id child-id)
(update-graph)              ;is called after a series of insert-node or remove-node calls. Inserts and
                            ;removes edges to make the graph consistent.
(apply-to-graph function)   ;applies a specified function to all nodes in the graph.
(dfs pre-visit post-visit)  ;applies pre-visit and post-visit functions to all nodes in the graph in
                            ;depth first order.
```

Class Graph-node

;Graph node wrapper around data structures stored in the graph class.

Slots:

```
:id           ;the id of the graph node, used for fast retrieval
:data         ;the actual data stored in the node
:parents      ;the ids of parents of the node
:children     ;the ids of children of the node
:pre-visit    ;used for cycle detection and graph traversals in specified order
:post-visit   ;used for cycle detection and graph traversals in specified order
:depth        ;the depth of the given node
```

Methods:

None

ITMS classes and data structures (web interface)

The web interface classes and data structures mimic the lisp data structures in Perl. The script lisp_to_perl.pl reads the ITMS data file into perl data structures which are then prompty serialized. Then, whenever the CGI scripts need certain information about the student or a career milestone, the appropriate file is read in. While the lisp application maintains one datafile, the web interface stores information for each student and each course and career milestone in its own file. This is done for efficiency and safety. .
We can ultimately expect large numbers of requests for this information, and we don't want to read in more information from file than necessary to satisfy a particular query (which will always be about a particular student, or a particular career milestone). Having to read in an entire datafile for each request would be prohibitively expensive.

ITMS server side authentication

ITMS uses the port of Apache web server for Windows. In order to let only authorized personnel access ITMS information on the web, server side authentication has been implemented. Authorization is granted and removed by means of two scripts, add_person.pl and remove_person.pl, respectively. The syntax is:

```
perl add_person.pl [name] [authorization_group] [password]
perl remove_person.pl [name] [authorization_group]
```

When the first script is ran an additional person is granted access with login [name], and password [password] to all information permitted to [authorization_group]. Some examples of authorization groups are Student, Course_author, Supervisor, Administrator.

Appendix B Phase I Prototype User Guide

The phase I implementation of the ITMS project consists of two heterogeneous parts: the GUI written in Allegro Common Lisp, and the web interface implemented as a series of Perl CGI scripts. The two parts interact through a file interface. Whenever ITMS generates a data file it gets converted to the format the Perl scripts can understand.
The ITMS GUI is meant to be ran periodically on the machine containing the web server and the CGI scripts. The GUI can be used to add and remove students, and edit the skill and career hierarchies. Furthermore, all e-mail communication currently goes through the GUI. The web interface is used to display useful information ITMS has collected and inferred on the web.

B.1 ITMS GUI Guide

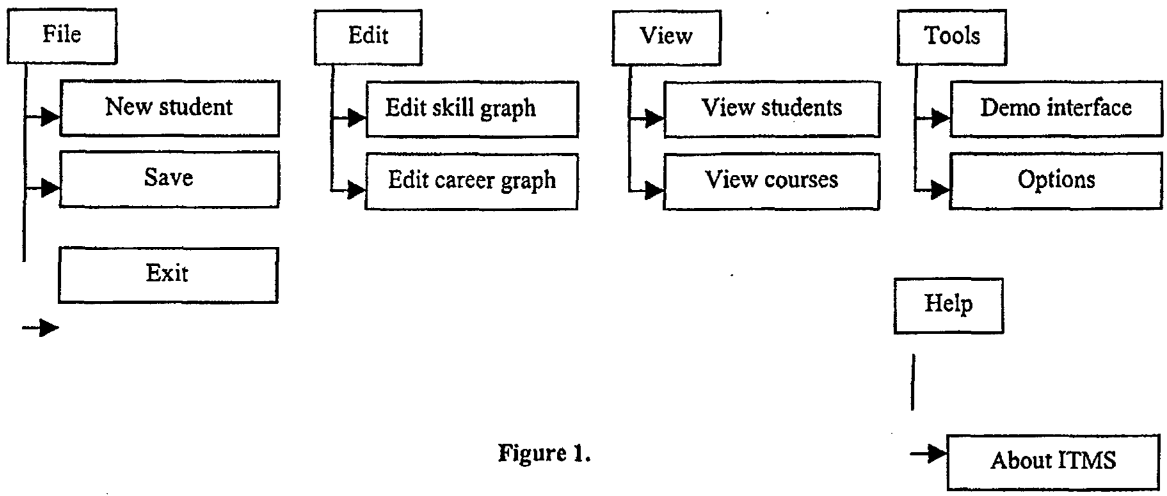


Figure 1.

Figure 1 shows the top-level choices available to a user of the ITMS GUI. Here is a short description of what each item does:

New student: Pops up the new student dialog box that allows the user to enter new student information, specify skills the student may have, etc.

The following buttons are available in the student dialog:

> **OK:** This finalizes the new student's settings and, if the user didn't forget to enter something, adds the student.
>
> **View skills:**
> **View career:**
> **Courses taken:** These three buttons view student information, however in the case of a new student this information is not yet available.
>
> **Cancel:** This cancels everything, and doesn't add the student.

Save: Saves the current state of ITMS into a datafile, and converts the file into a format understood by Perl CGI scripts.

Exit: Saves the current state of ITMS and exits from the GUI.

Edit skill graph: Pops up the skill graph editor dialog box that allows the user to modify the global ITMS skill graph (and edit individual skills). No conventional buttons are available in this editor, all actions are performed used the toolbar buttons. There are seven of these buttons and they look like this:

 **New node:** This pops up a new skill dialog. If the user doesn't cancel out of that dialog box, the new skill will be added.

 **Delete node:** This removes the currently selected skill (if any) from the global skill hierarchy.

 **Connect node to child:** One must have already selected a skill prior to clicking this button. Then the next skill you select will become a child of the currently selected skill.

 **Connect node to parent:** One must have already selected a skill prior to clicking this button. Then the next skill you select will become a parent of the currently selected skill.

 **Delete child:** One must have already selected a skill prior to clicking this button. Then the next skill you select will cease to be a child of the currently selected skill.

 **Delete parent:** One must have already selected a skill prior to clicking this button. Then the next skill you select will cease to be a parent of the currently selected skill.

 **Lose focus:** Pressing this button causes the editor to unfocus any skill that was previously selected.

Edit career graph: Pops up the career graph editor dialog box that allows the user to modify the global ITMS career graph (and edit individual career milestones). No conventional buttons are available in this editor, all actions are performed used the toolbar buttons. These buttons are identical to those found in the skill editor. The only changes are that the new node button will pop up a new career milestone dialog box, and all editor changes affect the career hierarchy, not the skill hierarchy.

View students: Pops up the search dialog that allows the user to search for individual students already in the system. This dialog shows the number of students currently in the system, a scroll-box that allows the user to select the criterion to search by (first name, last name, e-mail, etc)., a text box allowing the user to type their search key word, and a series of buttons:

Search: Returns a list of matches to the user's query.

New search: Resets the search, removes all matches.

OK: Commits all changes, and exits from this dialog box.

Delete: Deletes the selected match from the system.

Cancel: Is identical to OK in this context.

Furthermore, double-clicking on any match will bring up a dialog box showing information on the match (in this case a student dialog).

View courses: Pops up the search dialog that allows the user to search for individual courses already in the system. (Note: courses distinct from the career graph are deprecated).

Demo interface: Pops the demo interface dialog box that allows the user to move ITMS time backwards and forwards, and to receive 'e-mails.' The dialog box has three text fields that allow the user to specify the month, day, and year that ITMS considers to be 'today.' Furthermore, there is a text field where a user can enter a file name that will be read in by ITMS as an 'incoming e-mail.' There are also two buttons:

OK: This exits the demo interface dialog, and sets 'today's date' to be whatever the user last set in the date text fields.

Receive message: This 'receives an e-mail message' corresponding to the file specified by the user in the appropriate text field.

Options: Pops up the options dialog box that allows the user to modify e-mail settings (the server address, mail protocols, etc.) Not currently used, since not all of the e-mail functionality is fully in place.

About: Pops up the about dialog box. This box contains ITMS copyright information.

B. 2 ITMS Web Interface Guide

The main ITMS web page contains three buttons:

Login: This buttons triggers Apache server side authentication. In order to proceed, the user must provide a correct login and password. If the user succeeds in doing so, he will obtain credentials corresponding to the group his login is in for the duration of the browser session. If the user successfully logs in, he ends up in a page corresponding to the group his login is in. Let's assume the user is in the 'student' group. Then when he logs in, he will see a page displaying his name, contact information, etc. There will also be 5 buttons:

Refresh: This refreshes the page, committing any changes the user made to his information.

View skills: This takes the user to a page showing his skills in bar chart form. If the skill has children, then clicking on the bar corresponding to the skill will produce a different bar chart showing the sub skills. There are 2 buttons on this page:

Back: This takes the student back one level in his bar chart browsing. If the student as already seeing the bar chart corresponding to all the root nodes in the graph, he is taken to the student's page, but if not, he is taken to a page showing the bar chart containing the parent of the skills he currently sees.

Back to student's page: self-explanatory.

What's new: This takes the user to a page similar to the main 'what's new' page, the only difference being that only changes and updates to career milestones the student already accomplished are shown here.

Career counseling: This takes the user to a career counseling page that allows him to select career goals and get advice from ITMS. The page shows a scroll-list containing all career milestones the student accomplished, a text box where the user can enter the number of days he is allocating for achieving his career goal, and a list of yet-unaccomplished career goals. There are two buttons on this page:

Show plan: This button takes the student to a page showing the plan calculated to achieve his goal. The plan page always has the following two buttons:

Back to career counseling page: self-explanatory

Back to student's page: self-explanatory

Furthermore, if the plan contains career milestones which require some skill improvement on the part of the student, a button labeled 'find courses' appears next to each such career milestone. Then clicking that button will list all career milestones which will improve the student's skills to the required levels.

<u>Back to student's page:</u> self-explanatory.

<u>Back to main page:</u> self-explanatory.

<u>What's new:</u> This button takes the user to a page containing a list of hyperlinks corresponding to new career milestones. If the user attempts to follow a hyperlink here, he will have to get past Apache server side authentication (we do not want any person on the Internet to view career milestone information). The only button here is:

<u>Back to main page:</u> self-explanatory.

<u>Career milestone information:</u> This button takes the user to a page containing a list of hyperlinks corresponding to all career milestones currently in the system. If the user attempts to follow a hyperlink here, he will have to get past Apache server side authentication. The only button here is:

<u>Back to main page:</u> self-explanatory.

THIS PAGE INTENTIONALLY LEFT BLANK

DOCUMENT 3

Development Manual for 3D World Virtual Environment Software

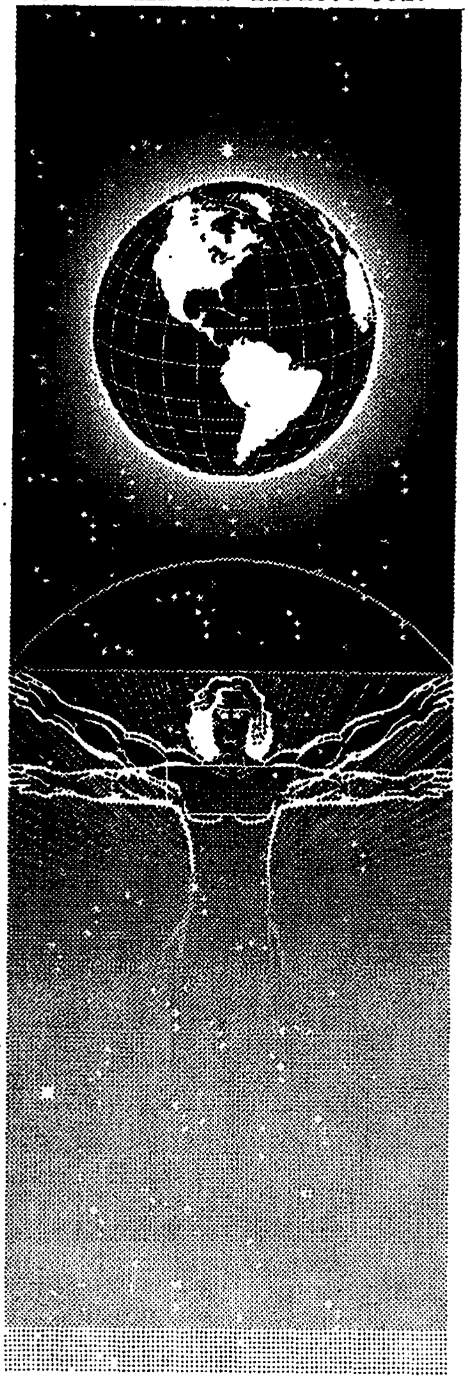
AD-A378892

December 1999

Sytronics Inc.
Dayton, OH

THIS PAGE INTENTIONALLY LEFT BLANK

AFRL-HE-WP-TR-2000-0017



UNITED STATES AIR FORCE RESEARCH LABORATORY

DEVELOPMENT MANUAL FOR 3D WORLD VIRTUAL ENVIRONMENT SOFTWARE

Annette L. McCoy

SYTRONICS, INC.
4433 DAYTON-XENIA ROAD
DAYTON OH 45432

Susan K. Schnipke

OHIO STATE UNIVERSITY
DEPARTMENT OF PSYCHOLOGY
COLUMBUS OH 43210

DECEMBER 1999

INTERIM REPORT FOR THE PERIOD APRIL 1998 TO NOVEMBER 1999

Approved for public release; distribution is unlimited.

Human Effectiveness Directorate
Crew System Interface Division
2255 H Street
Wright-Patterson AFB OH 45433-7022

NOTICES

TECHNICAL REVIEW AND APPROVAL

AFRL-HE-WP-TR-2000-0017

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

MARIS M. VIKMANIS, DR-IV
Chief, Crew System Interface Division
Human Effectiveness Directorate
Air Force Research Laboratory

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1999	Interim, April 1998 to November 1999

4. TITLE AND SUBTITLE

Development Manual for 3D World Virtual Environment Software

5. FUNDING NUMBERS

C: F41624-94-D-6000
PE: 62202F
PR: 7184
TA: 14
WU: 25

6. AUTHOR(S)

Annette L. McCoy*
Susan K. Schnipke**

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

*Sytronics, Inc.
4433 Dayton-Xenia Rd, Dayton OH 45432

** Ohio State University
Department of Psychology, Columbus OH 43210

8. PERFORMING ORGANIZATION

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory
Human Effectiveness Directorate
Crew System Interface Division
Air Force Materiel Command
Wright-Patterson AFB OH 45433-7022

10. SPONSORING/MONITORING

AFRL-HE-WP-TR-2000-0017

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report documents a software package called **3DWorld**. The software provides the environment and scenario development tools necessary to create a virtual environment for human performance research. This report contains step-by-step instructions on how to develop and run virtual environments, as well as an in-depth description of the program structure.

14. SUBJECT TERMS
Virtual Environments Human Performance

15. NUMBER OF PAGES
60

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNLIMITED

NSN 7540-01-280-5500

Standard Form 298 (Rev 2-89) Prescribed by ANSI Std Z-39-18
298-102 COMPUTER GENERATED

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

1.0 INTRODUCTION

3D World is a virtual environment software package created by scientists in Air Force Research Laboratory (AFRL) at Wright-Patterson Air Force Base, Ohio. The program, which runs under MS-DOS, allows users to design virtual environments, customize scenarios, navigate within the environments, and collect experimental data.

Gerald Dalley, a summer intern with AFRL, developed the original version of 3D World primarily for studying situation awareness issues. A general definition of situation awareness is "the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future," (Endsley, 1993). Using 3D World environments, we can study situation awareness by researching how people perceive their surroundings, navigate within those surroundings, and remember locations of objects. To date, 14 situation awareness studies have been conducted using 3D World environments. Results of six of the studies have been published (Colle & Reid, 1998; Colle & Reid, 1999), and the others are a series of studies which are near completion. In addition, 3D World environments were also used to study workload issues at Ohio State University (Nygren, Schnipke, & Reid, 1997) for which the environments were customized to measure time pressure, effort, and stress.

In this paper, we will be explaining how to build an environment, how to view and navigate within the environment, how to customize scripts or scenarios, and how to collect data while running the program. First, read the overview which will provide you with general information about the program and give you a better sense of what the 3D World program has to offer. Then go on to the more detailed sections of the manual for in-depth information about creating environments.

2.0 OVERVIEW

Building an environment in 3D World can be very simple or very complex, depending upon what you want. You may want a simple world which consists of a small building with a couple of rooms in which you are free to roam around, or you may want a multi-level, multi-room environment, equipped with stairs and elevators, and monitored movement. We will briefly overview what is involved in building a small, basic environment so you'll have an idea of what to expect.

To begin, take a look at the following illustrations to get an idea of what an environment may look like on the screen. Figures 1, 2, and 3 are a sequence of captured screen images as the operator is walking down a hallway.

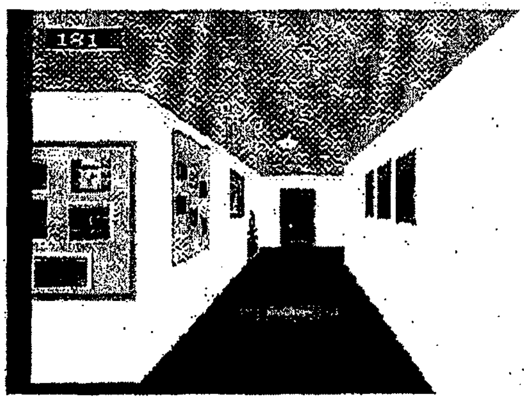


Figure 1

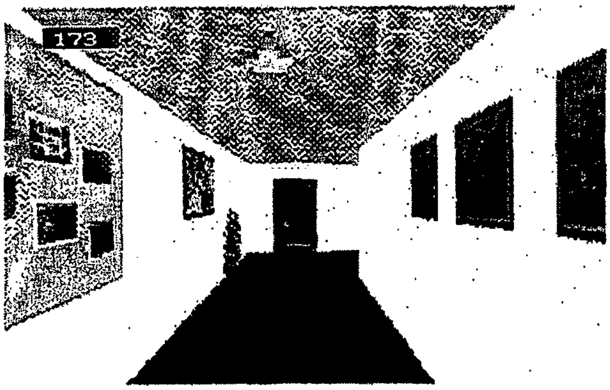


Figure 2

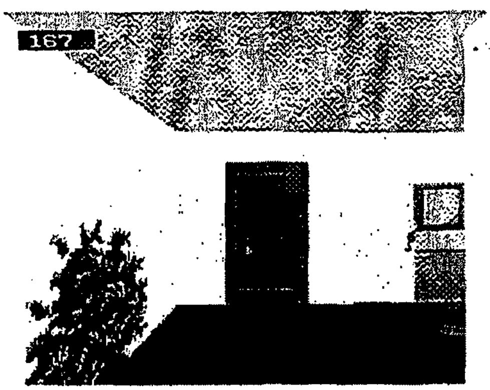


Figure 3

Figures 4 through 10 are various captured screen images of environments used in situation awareness (Colle, Reid, 1997) and workload (Nygren, Schnipke, 1997) studies.

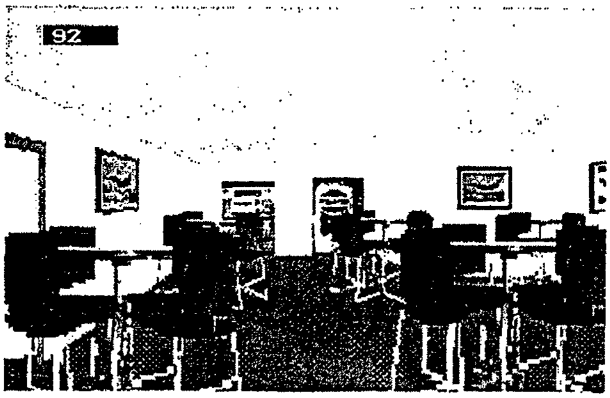


Figure 4

Figure 5

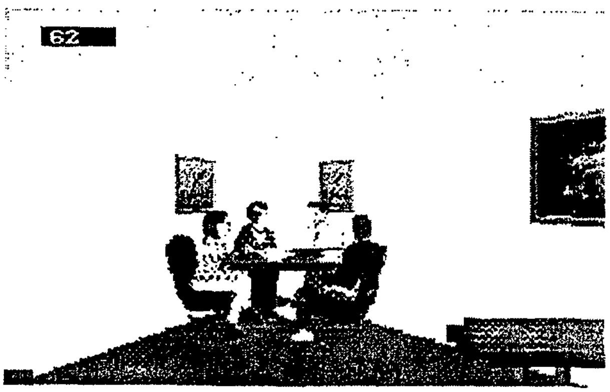


Figure 6

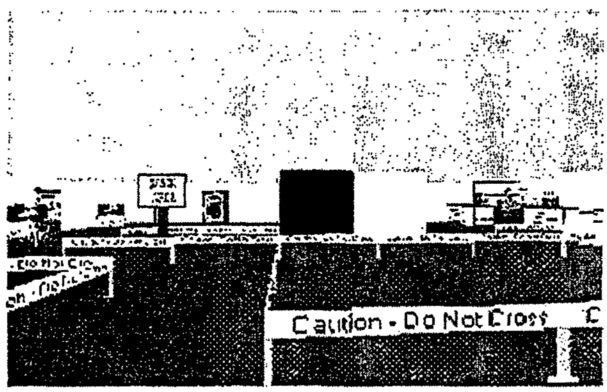


Figure 7

Figure 8

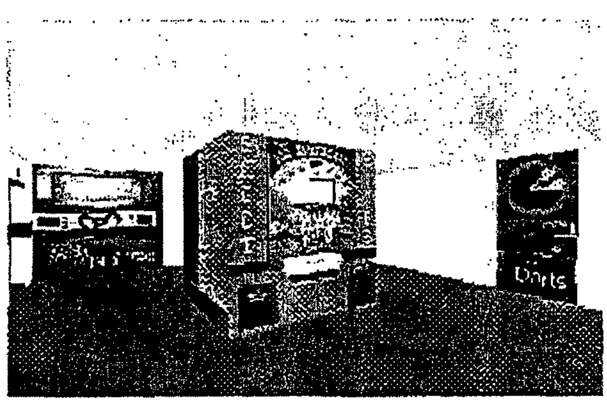


Figure 9

Figures 1-9 are examples of what you would see while 'viewing' the environment. The environment is comprised of several pictures called image files. Image files are typically .pcx files created in a paint program. Each picture (image file) represents a wall or an object in a room. For example, one picture could be a plain wall. If you organized pictures of a plain wall in the form of a square, you'd have a square room. To add a door, you would include a picture of a wall with a door on it, or you could add a wall with a mirror or window, etc. Organize these pictures to form rooms and hallways and you create an environment. All of these image files are gathered into one of two *Icon Description* files: mapdata.def or objectdata.def.. *Mapdata.def* contains the image files which represent *walls* of the enviroment. *Objectdata.def* contains images which represent *objects* within rooms, such as a chair. Both of these files are displayed on a drawing board called the *Map Editor*. The *Map Editor* is the tool used to build or create the environment. See Figure 10.

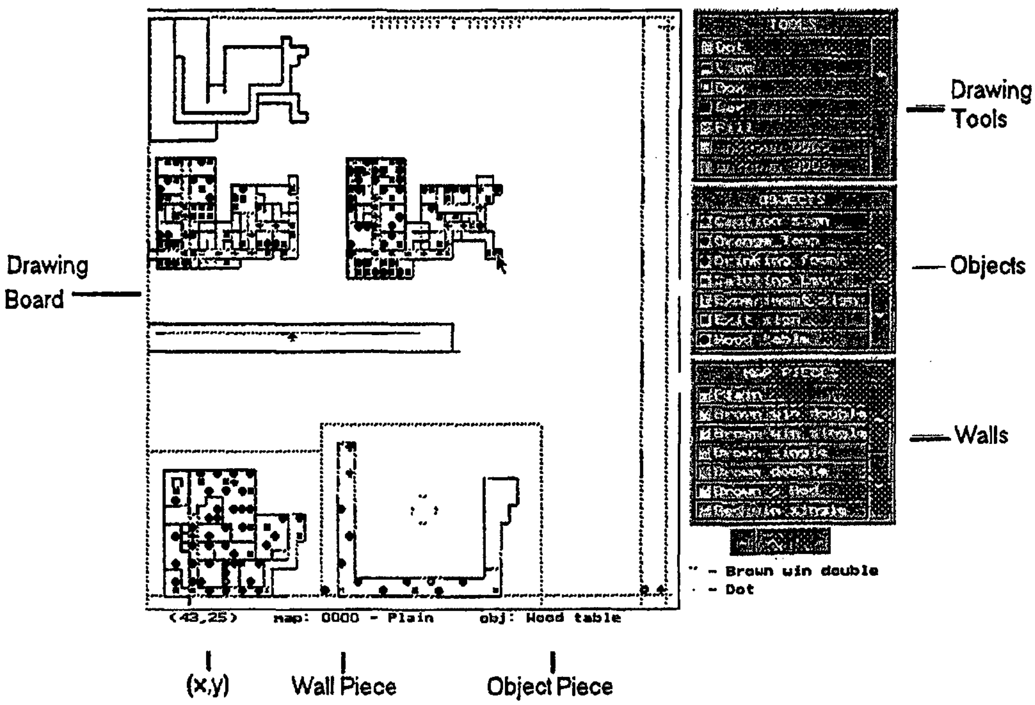


Figure 10. Map Editor drawing board

As you see in Figure10, the object and wall pieces are listed to the right of the actual drawing area. To create an environment, you simply select a map piece by clicking on it with the mouse, then click on the drawing board where you want to place it (more detailed instructions will be provided later). For example, the above drawing area shows five individual environment segments, which can represent different levels of one building, or different environments all-together. It also shows a long corridor which appears about midway down the drawing board. The line segments on the board represent the walls of the environment and the circles, squares, etc. represent objects within the rooms and hallways. Listed in the object icon menu, there are also startpoint arrows to choose from which you select and place at the position you want to start operator control in the environment. Wherever the arrow is placed is where you will begin viewing the environment when running the 3D program. More information is provided on the Map Editor in Section 4.2.

To briefly summarize, to create and operate in a virtual environment, you 1) collect multiple image files which represent walls and objects you want in your environment (See Section 3.1), 2) list all images in the Icon Description files to be displayed on the Map Editor, 3) build the environment using the Map Editor drawing board, 4) create a scenario, 5) run the 3D program, and 6) navigate in the environment and collect data.

Of course, this is a very simplified overview. More detailed information is provided throughout the manual. For questions regarding this document or the 3D World Program, see the references at the end of the manual.

*Note: For the remainder of this manual, the term "user" refers to the person using the 3D World software to develop the environments; "operator" refers to the person who is navigating in the environment in the run mode.

3.0 UNDERSTANDING THE FILES YOU WILL USE

3D World requires that the following files be present within a directory in order to create a working environment. These files can be categorized into five different groups:

1) **Image Files** (.pcx files)

2) **Icon Description Files**
    a) objdata.def
    b) mapdata.def
    c) tools.def

3) **World Database File** (3d.map)

4) **World Development Files**
    a) editmap.exe
    b) editor.exe
    c) initmap.exe

5) **Loading and Running Files**

a) 1.way
b) egavga.bgi
c) map.3dm
d) 3d.exe

3.1 Image Files

In order to create an environment, you need image files which portray walls, doors, objects, etc. 3D World uses .pcx image files created with an independent paint program. All .pcx files must be 128 x 128 resolution, 256-color .pcx graphics files. In general, the same color palette should be used for all images. The maximum number of .pcx files you may use in any single environment is 254.

3.2 Icon Description Files

The Icon Description files define the map piece icons and drawing tools which will be used to build the environment on the Map Editor drawing board (see overview). There are three different Icon Description files: 1) mapdata.def, 2) objdata.def, and 3) tools.def. The mapdata.def and objdata.def files specify the name, color, and shape of the wall and object map pieces (respectively) to be used in the Map Editor. The Tools.def file defines the tools which are available to assist in building the environment. The format is similar for all three files.

3.2.1 Mapdata.def

An example of a mapdata.def file is as follows:

0000 eeff plain
0001 7aff window
0002 66cc door
0003 29ff mirror
0004 11ff clock
00FF 00ff empty

*Note: The following line should ALWAYS appear at the end of the mapdata.def file:

00FF 00FF Empty

The first field of numbers represents the order of the item in the file. The second field defines that item's icon image. The icon will appear as two lines on the drawing board map. The first two positions in the second field define the color of the icon; the last two positions define the bit patterns for those lines (for more information on how bit patterns work, see setlinestyle documentation in either Borland Pascal or Turbo C manuals).

The line colors are defined as follows:

0 – Black	8 - Dark Gray
1 – Blue	9 - Light Blue
2 – Green	A - Light Green
3 – Cyan	B - Light Cyan
4 – Red	C - Light Red
5 – Magenta	D - Light Magenta
6 – Brown	E - Yellow
7 - Light Gray	F - White

The third field is the descriptive name for the wall icon. The descriptive name does not need to be the same as the .pcx file it represents. The icon and its descriptive name will appear in the Map Editor and is for identification purposes only.

Using the example mapdata.def file above, the mirror is 0003 (the fourth line) on the list. (*Note - the actual name of the mirror .pcx file **must also be listed fourth** in the 3d.map file discussed in Section 3.3). The icon that will appear on the map to represent this mirrored wall will be a pair of lines, one green (#2) and one light blue (#9).

The first and second fields are all hexadecimal numbers (all digits 0..f) and **MUST** be four characters long. Each field **MUST** be separated by one and only one space. Additionally, there **MUST** be **NO** blank lines or lines that do not follow the above conventions.

3.2.2 Objdata.def

There are two differences between mapdata.def and objdata.def: 1) specific starting point lines must be included in objdata.def, and 2) the images are presented as objects rather than walls in the environment. An example of an objdata.def file follows (the required starting point lines are in bold):

0000 1056 Box
0001 60f0 Tree
0002 b016 Chair
0003 5120 Person
00F7 20F0 Starting point
00F8 20F1 Starting point
00F9 20F2 Starting point
00FA 20F3 Starting point
00FB 20F4 Starting point
00FC 20F5 Starting point
00FD 20F6 Starting point

00FE 20F7 Starting point
00FF 0010 Empty
The object icons will appear on the map as symbols such as a circle or square. Here is the key for icons in Objdata.def:

1st position
0-F (see above): foreground color

2nd position
0-F (see above): background color

3rd position
0 - print character in 4th position
1 - solid
2 - half-tone
3 - solid w/decoration
4 - half-tone w/decoration
5 - circle
6 - horizontal door
7 - vertical door
8 - top half foreground, bottom back
9 - dot
A - tall upper left
B - short upper left
C - centered
D - x **(associated with 4th position)**
  0 - no background
  1 - show background bar
  2-f - reserved
E - outline
F - arrow (direction determined by 4th position)

4th Position
When D is in 3rd position:
  0 - no background
  1 - show background bar
  2-f - reserved
When F is in 3rd position:
  0 - north
  1 - northeast
  2 - east
  3 - southeast
  4 - south
  5 - southwest
  6 - west

7 – northwest
*Note: If D or F is not in 3rd position, the 4th position is ignored, but a number must be inserted.

3.2.3 Tools.def

The Tools.def file lists the 'drawing tools' and defines the shape and color of their corresponding icons which appear in the Map Editor. The "toolbar" is similar to those used in independent paint programs. **YOU SHOULD NOT EDIT THE TOOLS.DEF FILE.** There will be more information regarding the tools in Section 4.2.2.1.

3.3 World Database File (3d.map)

3d.map is considered to be the world's database. It calls up and defines all image files as walls or objects, it calls up the environment map, and it defines walking parameters, turnrate, and other miscellaneous parameters. It defines any overhead images (to be discussed later in 3.3.5) and Help screens to be used in the environment. There are six major sections to the 3d.map file: 1) map - defines the map for the environment, 2) parameters - defines the walking parameters, 3) pic - defines the wall images, 4) obj - defines the object images, 5) overhead - defines the overhead images, and 6) help - defines the help screens. Here is an example of an entire 3d.map file:

```
[map]
map.3dm

[parameters]
stepHeight .50
eyeLevel 5.0
speed 31
stepDist 3
turnRate 100

[pic]
wall1.pcx
basewall.pcx
poster.pcx
sign.pcx
window.pcx window
door.pcx door
windoor.pcx door window
opendoor.pcx opendoor window

[obj]
table.pcx
```

[overhead]
map.pcx
Cafeteria
Conference

[help]
help.pcx

3.3.1 [map] section

The [map] section calls up the map of the environment which is a file named map.3dm. This is the map that is created and edited in the Map Editor and it is **always saved** as map.3dm. Only the map.3dm file can be loaded in this section, therefore the following lines **must** appear in the 3d.map file:

[map]
map.3dm

If you would like to have multiple versions of maps, simply rename other versions (i.e., mall.3dm or kitchen.3dm) and keep them in the 3D directory. Just remember that the map you want to view in the map editor must be called map.3dm.

3.3.2 [parameters] section

This section sets up the walking parameters for 3D World. The **stepheight** controls the "bobbing" sensation when walking. If this value is increased, the "bobbing" sensation is increased. The **eyelevel** is the height of the field of view. In **USEKEYS** mode, the **speed** dictates how many "steps" it will take to cross a coordinate block of the map, which represents 8 square feet. The **stepdist** should be an estimate of how long the step is, however, changing the value does not seem to change the actual step distance. The **turnrate** is how many key presses it takes to move a certain number of degrees in a circle. Step height, distance, and eye level are measured in units of feet, walking speed in miles per hour, and turning rate in degrees per second. These units will vary according to the computer you are using and how the waypoint file is set up to navigate; i.e., in USEKEYS or USEZOOM mode (see 3.5.1.1). Using a 486 DX-4, 100mhz computer with an Intel processor, and the USEKEYS mode set in the waypoint command file, the following parameters "seem natural":

[parameters]
stepheight .50
eyelevel 5.0
speed 31
stepdist 3
turnrate 100

You may change any of these values to suit your needs. Remember, the values are highly dependent upon the computer you are using so there are no set values defined. If no values are entered or if any values exceed the 8-foot boundaries of a coordinate block, 3D World will give a short error message telling you what the problem is. The step distance must be entered after the speed, and if any values are changed, any .mov files (see 3.5.1.5) that are being used must be rerecorded. Here are some examples of how changing the above parameters effect the keystrokes in the **USEKEYS** mode:

Parameter	Approximate Number of Keystrokes
turnRate 100	≈ 71 keystrokes for a complete circle. ≈ 5° per keystroke.
turnRate 200	≈ 40 keystrokes for a complete circle. ≈ 9° per keystroke.
speed 31	3 blocks ≈ 6 keystrokes. 4 blocks ≈ 11 keystrokes. 7 blocks ≈ 16 keystrokes.
speed 12	3 blocks ≈ 13 keystrokes. 4 blocks ≈ 17 keystrokes. 7 blocks ≈ 45 keystrokes.

Figure 11. Examples of changing parameters in the USEKEYS **mode**

Changing parameters in the **USEZOOM** mode will not show significant differences. Although all parameters must be entered, the important parameters are the stepHeight and the eyeLevel in this mode.

3.3.3 [pic] section

The [pic] section defines the image (.pcx) files to be used for the environment's walls. **These files must be in the same order as the items in the mapdata.def file (section 3.2.1)** in order for the world editor to accurately represent the environment. The descriptive name in the mapdata.def file need not be identical to the actual .pcx filename, but the order should be exact. Here is an example of the [pic] section containing three image files.

```
[pic]
bluewall.pcx
bathroom.pcx
officwin.pcx
```

There are three features that can be added to a basic wall. The first feature is to make part of the wall piece **transparent**. The second feature is to have a **door** that can be opened and walked through. The last feature is to have an **open door** which can be walked through.

3.3.3.1 Partial Transparency

This feature allows part of the wall to be transparent. This is useful for making windows and open doorways. The part of the image file that should be transparent must be black. The .pcx filename must be followed by the word window in the 3d.map file to take on transparent qualities. For example, the image file officwin.pcx listed above refers to an office window. To make the black areas appear as a window, you must type the word window following the filename.

Example:
officwin.pcx window

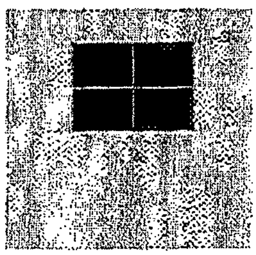


Figure 12. Officwin.pcx as it appears in a paint program

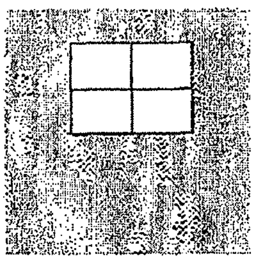


Figure 13. Officwin.pcx as seen as a wall in the Environment.

Wall.pcx would appear as a wall with a transparent window in the environment. Anything on the other side of the wall would be visible through the window.

3.3.3.2 Door

Normally you may not walk through a wall, however, the "door" feature of a wall provides the ability to pass through the wall piece. The .pcx filename must be followed by the word door in the 3d.map file. To pass through the door, the spacebar must be pressed when the operator is next to the door. Generally, this is used for closed doors so that when the operator walks up to the door, they must stop and press the spacebar to get in, which would simulate stopping to open the door.

Example:
bathroom.pcx door

3.3.3.3 Open Door

The "open door" feature allows for passage through a wall piece without pressing the spacebar. Often, this feature is used with the "window" feature so the perception is a door standing open that can be looked through and walked through.

Example:
bathroom.pcx opendoor window

Examples as seen in a Paint Program:

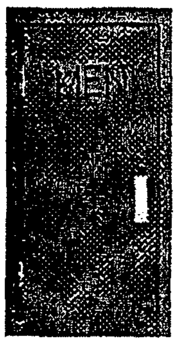


Figure 14 Bathroom.pcx door

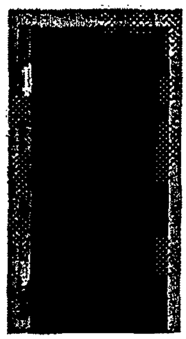


Figure 15 Bathroom.pcx opendoor window

In the environment, if standing in front of Figure 13, you could press the spacebar and be placed on the other side of the door as if you had walked through the doorway. In the environment, the black area in Figure 14 would appear transparent and you could walk through the doorway.

3.3.4 [obj] section

The [obj] section defines the .pcx files to be used as objects in the environment. Objects are placed within rooms, and they do not hinder movement when navigating through an environment. One coordinate block on the map is considered to be an 8 x 8 foot room, and only one object can be placed 'within' a coordinate block. An object is actually flat, considering that it is an image file like the walls. Therefore, it only has one view, so no matter what angle it is viewed from, it will always appear the same. An example is that if the image file of a person facing you is placed in a room as an object, the person will always appear to be facing you regardless of the viewing angle.

In summary: 1) objects are placed within a coordinate block on the map whereas walls define the perimeter of coordinate blocks, 2) the space surrounding the object in the image file should be painted black in order to appear transparent in the environment, 3) you can walk through objects, but not walls (unless designated as open doors). Note: If the space surrounding the object is not painted black, the object will appear to be a wall that you can walk through.

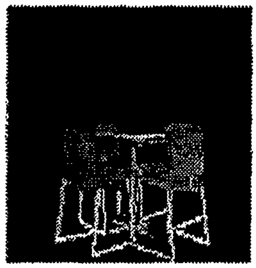


Figure 16. Table.pcx as it appears in paint program.

Figure 17. Table.pcx as seen as an object in the environment.

3.3.5 [overhead] section

This section gives information for displaying an 'overhead' picture in the environment. An overhead picture is an image file which can have movable text pieces.

The text pieces are originally displayed as white text listed down the left side of the image file. Each piece of text can be moved around within the image file using the drag and drop feature of a mouse. The first entry in the [overhead] section is the file name of the .pcx file to be displayed. All successive lines define the text pieces to be listed. In the following example, a map (map.pcx) will be loaded as an overhead image file, and Office, Cafeteria, Files, and Conference will be the movable text pieces.

```
[overhead]
map.pcx
Office
Cafeteria
Files
Conference
```

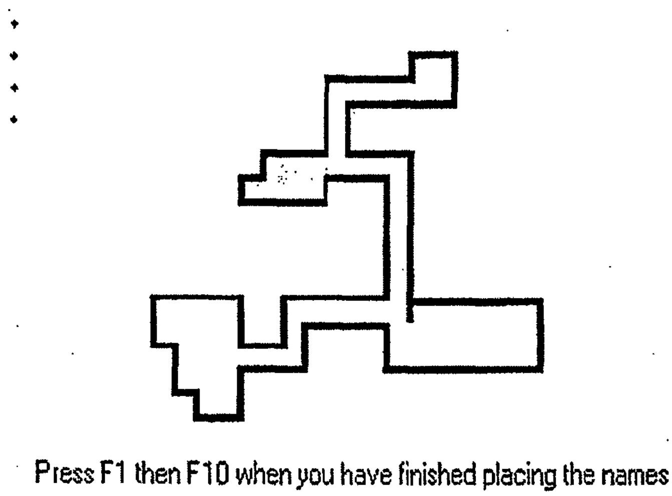


Figure 18. Overhead Image of a map

In this case, the user would select the text pieces with the mouse and move them to the desired location on the map. Another example of an overhead image would be a schedule like the one shown below.

[overhead]
schedule.pcx
Annette
Susan

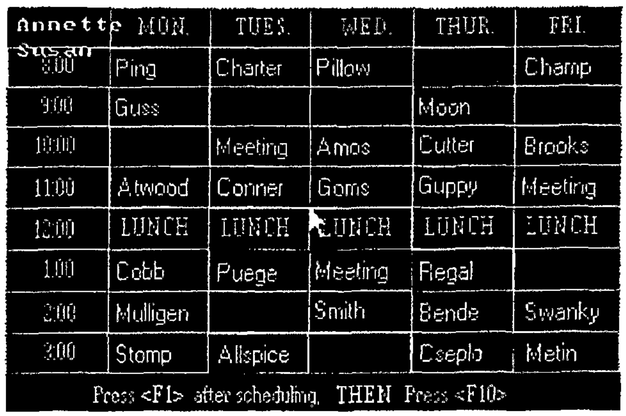


Figure 19. Schedule.pcx as an Overhead Image.

In this example, the two names in the top left corner, Annette and Susan are the moveable text pieces, and they can be selected and moved to the appropriate cell.

There are two ways to call up an overhead picture: 1) calling it up in the waypoint command file using the SHOWMAP command (Section 3.5.1.6), or 2) showing the picture on demand with the key combination of Alt-m.

To exit the overhead picture, <F10> must be pressed. If the picture is shown more than once, the pieces will appear wherever they were left in the previous display; however, the position is **not** saved in the data output file. Currently, the only known method to save the final positions of the names, is to run a screen-capturing program simultaneously with 3D World.

3.3.6 [help] section

This section is a one-line entry that displays a customized help screen when the key combination of Alt-h is pressed. The help screen is a .pcx image file no larger than 360 by 200 pixels. To exit the "help" screen, press any key.

[help]
help.pcx

3.4 World Development Executable Files

The World Development files are the World Editor and the Map Editor. These are the executable files that are used to actually build the environment. The World Editor is the primary interface between the user and environment, and the Map Editor is the drawing tool used to construct the environment. You will be instructed how to use the editors in Section 4.0.

3.4.1 Editor.exe

Executes the World Editor Menu System (Section 4.1).

3.4.2 Editmap.exe

Executes the Map Editor which is the drawing board and tools for building the environment (Section 4.2).

3.4.3 Initmap.exe

Executes the Map Editor, but creates a blank drawing board (Section 4.2).

3.5 Loading and Running Files

The following are files that are needed to run the 3D World program. The waypoint command file is the primary running file in that it is within the waypoint file (1.way) that you can define a scenario or story for the environment, designate sound files, provide instructions or define routes within the environment, etc.

3.5.1 The Waypoint Command File (1.way)

The waypoint command file is the central running file for 3D World. It is responsible for loading the map, initializing sound, providing interactive information, controlling all action in the environment by designating waypoints (positional goals), and defining data collecting procedures.

A positional goal is a coordinate block (waypoint) within the map of the environment that the operator 'triggers' once it is stepped on. For example, an operator may be asked to go to a particular room in an environment and find a specific object. When the operator walks up to the object, that waypoint may trigger, and he may get a text message on the screen and/or a voice message confirming that he has found the object. A waypoint is simply a place where instructions are provided, voice files are activated, overhead images are displayed, or some other action may occur.

The waypoint file is created using any ASCII text editor. The 1.way file is created and/or edited in the DOS editor by typing "edit 1.way" at the DOS prompt. It can also be created and/or edited in Windows using Notepad or any other editor. Regardless of the editor used, the file should always be saved as text with .way as the extension. You may have multiple waypoint files in a directory, for example. 1.way, 2.way, 3.way, etc. 3D World only recognizes 1.way, however, unless you initiate the executable with the w = somenumber command when you start the program (See 3.2 for more information).

A waypoint file is partitioned into sections called functions. A function begins with a header (usually waypoint coordinates) and lists a set of commands to be carried out when that function is called. The first function in the waypoint file, however, is unique in that it provides instructions to the run module for setting up the environment.

3.5.1.1 The Start Function

The headers for all functions are enclosed in square brackets. The first function always has a "start" header rather than waypoint coordinates. Following the start header, commands are given to set up the environment. Typically, the commands in the [start] function include: 1) display a picture while the map is being loaded, 2) load the world database .map file, 3) activate the sound, 4) activate the input device to be used and 5) activate the first waypoint. Here is an example of the [start] function.

```
[start]
showpalpic logo.pcx
load 3d.map
usesoundeffects
usedigitizedvoice
backgroundsoundoff
usekeys
wait
activate (17,6,1)
play
```

The following is a list with explanations of the commands that can be used in the [start] function.

• SHOWPALPIC

Displays a picture centered on the screen while the .map file is being loaded. Typically, an introduction screen is displayed. The picture's palette is also being loaded at this time. In general, the same palette should be used for all images. If multiple palettes must be used, the palette can be reset by using the SHOWPALPIC command.

• LOAD

Loads the 3d.map file. This command is essential to run 3D World. The 3d.map file is the world database and calls up the images for the environment. Only one LOAD should be performed per waypoint file and the LOAD command should always precede a PLAY command. The file does not have to be called 3d.map, but does have to have a .map extension.

• USESOUNDEFFECTS
Enables the playing of sound effects to signal running into walls and walking through doors. If this option is turned on, ouch.voc and phaser8.voc must be present in the current directory.

• USEDIGITIZEDVOICE
Enables the playing of digitized sound files with the **SPEAK/SPEAKONLY** commands. This allows you to hear voice files of people talking.

• BACKGROUNDSOUNDON/ BACKGROUNDSOUNDOFF
BACKGROUNDSOUNDON turns on background sound. It causes **test.mid** to be played continuously, until the program exits, or BACKGROUNDSOUNDOFF is executed which simply turns off the background sound started by BACKGROUNDSOUNDON. Successive calls to BACKGROUNDSOUNDON will simply restart the background music.

Note*: To hear sound, the /sound command must be present when starting 3D World (see Starting the Program, Section 3.1). These features have only been successfully tested using true SoundBlaster cards.

• USEKEYS / USEZOOM
To navigate in an environment, the keyboard or the mouse can be used. The default navigation device is the keyboard. These commands specify the navigation device. None of the keyboard commands have associated parameters, therefore each should be entered on a line by itself.

USEKEYS changes the 3D viewer's input to the standard keyboard input. For slow machines like a 386DX/20, this mode is recommended since keystrokes will not be "missed" if the computer samples at the wrong time. The keyboard buffer is reduced down to one stroke, and additional keystrokes are ignored until the first is processed. This navigation device is also recommended when the feeling of "stepping" is desired.

USEZOOM is a faster implementation of USEKEYS. The "zoom" mode gives more of an arcade game feel, but requires more processor speed. If the processor is too slow and can't provide a reasonably high update rate, users may notice quick keystrokes being missed by the computer. This mode allows multiple keystrokes to be pressed for turns while walking, etc. The motion also tends to seem more fluid, and thus the sensation of "stepping" is reduced.

• USEMOUSE / NOMOUSE

USEMOUSE enables the mouse as a navigation device. This command must be used in conjunction with either USEKEYS or USEZOOM. The mouse is generally not used as a navigation device because it is difficult to control.

NOMOUSE disables the mouse as a navigation device.

• WAIT

The WAIT command used at the end of the first function will keep the image file that was loaded using the SHOWPALPIC command on the screen until the operator presses a key. A message reading "Press a key to continue" will blink on the bottom of the screen. Once a key is pressed, the environment will appear on the screen at the starting point. If this command is not used, the image file will only be on the screen long enough for the 3d.map file to be loaded, then will disappear and the environment will appear at the starting point. This usually happens vey quickly, so it is recommended to use the WAIT command in the start function.

• ACTIVATE / DEACTIVATE

When in play mode, the 3D World viewer checks to see if an active waypoint block has been intersected (stepped on) each time a movement takes place. When an active waypoint has been triggered, execution goes to the commands associated with that coordinate block.

The first waypoint should be activated in the [start] function. ACTIVATE activates a waypoint. The coordinates in (x,y,l) format of the waypoint should follow the word activate. The x and y are standard coordinates and the 'l' identifies the level of the waypoint. You may have multiple levels of a waypoint if you want it to do different things at different times.

When the ACTIVATE command is called, a search is conducted within the entire waypoint file to find the header containing that waypoint. Using the ACTIVATE example shown in the [start] function above, a line beginning with "[(17,6,1)]" would be searched for. If found, the commands associated with that waypoint would be executed.

DEACTIVATE simply deactivates a waypoint so it will not trigger if stepped on again. This allows the operator to visit a location more then once, but only have an action occur at the time that the waypoint is active. This command is essential when using multiple level waypoints. If a waypoint is deactivated within it's own function, DEACTIVATE MUST BE the last command before PLAY.

• **PLAY / END**

The final command in a function is either PLAY or END. Either of these commands signals the end of the function. PLAY is the command used at the end of all functions to return control to the operator. While in play mode, navigation of the environment is possible until a waypoint is stepped on. At that time, control is given to the function associated with that waypoint. END is used in the last function to end the 3D viewing program.

3.5.1.2 Waypoint Functions

All functions following the "start" function act as positional goals called waypoints. Each waypoint function contains a header which indicates the coordinates that activate the waypoint. After the waypoint is activated, commands following the header are executed. The waypoint is terminated when a **PLAY** or **END** command is encountered.

The header of every waypoint **must** be enclosed in brackets. The coordinates for a waypoint are in the (x,y,l) format. The x and y are standard coordinates and the 'l' is for the level of the waypoint. Each waypoint can have different levels so that each time the waypoint is crossed the program will display different information. The levels are numbered consecutively beginning with one. An example header would appear like this:

[(17,6,1)]   x = 17, y = 6, Level = 1

There are a variety of commands which can be executed in a waypoint function. Here is an example of a function in a waypoint file.

```
[(17,6,1)]
record effort.mov
speak intro1.voc
echo "Please raise your hand and
echo "wait for the experimenter
echo "before beginning.
showmap schedule.pcx
showpic blank.pcx
speakonly 16min.voc
nop
speak intro2.voc
echo "You've just arrived at the
echo "Browning, Browning & Smith
echo "law firm. Enter the office to
echo "find out what you will be
echo "doing today.
clock_on 500
activate (15,15,1)
```

deactivate (17,6,1)
play

There are many different commands that can be used in the waypoint file. They have been categorized into four groups: 1) basic functions, 2) interactive commands, 3) data collection commands, and 4) overhead image commands. The various commands are defined below. The command interpreter is not case sensitive, however it is advised that all commands be lower case. Each command must start on a new line to be interpreted correctly, and each command must end with a hard return. Extra spaces are ignored.

3.5.1.3 Basic Commands

• **ACTIVATE** activates a waypoint. The 3d-world viewer checks to see if a waypoint coordinate block has been activated each time a movement takes place. When an activated waypoint is intersected, a search is conducted throughout the entire waypoint file to find the function header containing that waypoint. When in play mode, the execution goes to the first line following the header.

Example:
activate (17,6,1)
activates the function [(17,6,1)] shown above

• **DEACTIVATE** deactivates a waypoint. It is important to deactivate a waypoint once it's been activated if you do not want it to trigger again. By deactivating a waypoint, that point is removed from the list of waypoints to be checked against as described in **ACTIVATE**. If a waypoint is reached, but not deactivated before a **PLAY** command is executed, that waypoint will continue to trigger.

Example:
deactivate (17,6,1)

• **END** ends the 3D viewing program. The output file is closed, cleanup is performed, and control is returned to the DOS prompt.

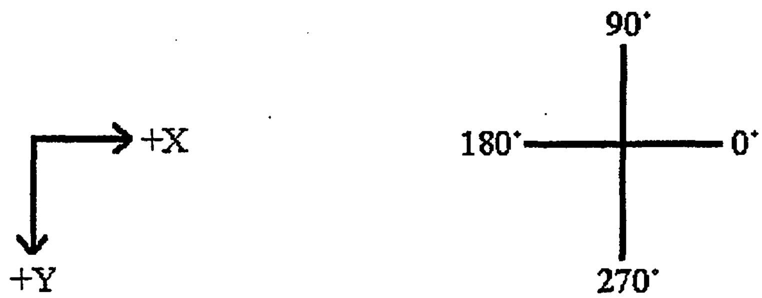
• **NOP** is short for No Operation. This is good for separating dialogue boxes when you have long segments of text.

• **PAUSE** stops the program for a specified amount of time. This command requires a parameter representing the number of seconds for the program to pause. This command can be used in conjunction with the **SHOWPIC** command to create a sense of animation.

Example:
pause 1

• **PLAY** returns control to the viewing program. While in play mode, navigation of the environment is possible until a waypoint is hit. At that time, control is given to the waypoint function associated with that point. A **PLAY** command will almost always be at the end of a waypoint function.

• **POSITION** teleports the operator to a specified location. The values given are the X and Y coordinates and the facing angle. The point (0,0) corresponds to the upper left-hand corner of the world. The X position moves positively to the right and the Y position moves positively downward (the standard quadrant system uses up as positive Y). The angle is measured in degrees where +X is 0°, -Y is 90°, -X is 180°, and +Y is 270°. Five squares to the right, ten squares down from the upper left-hand corner facing up would be (5,10,90).



Example:
position (5,10,90)

Figure 20. (X,Y) Coordinate System and Angles.

This is a useful command when it is necessary to take the subject out of one environment and put them in another, or position them on the other side of a door.

• **POSITIONWAIT** is identical in syntax to the **POSITION** command. The difference is that 3D World waits for a key to be pressed before continuing.

Example:
positionwait (5,10,90)

• **REM** is a remark statement or comment. Any line beginning with this is for internal documentation and is ignored by the run module. Additionally, blank lines and invalid commands are treated as comments.

• **WAIT** pauses until a key is pressed. It also writes a blinking message of "Press a key to continue" centered on the bottom of the screen.

• **WAITFORENTER** pauses until the <Enter> key is pressed. It also writes a blinking message of "Press the <Enter> key to continue" centered on the bottom of the screen.

• **YO** pauses until a key is pressed. The difference between this and the **WAIT** and **WAIFORENTER** commands is that you can customize a message to appear on the screen. The message can be up to 40 characters in length. The syntax is the same as the **ECHO** command (described below).

Example:
yo "<user defined message>

• **YOWAITFORENTER** pauses until the <ENTER> key is pressed. The difference between this and the **WAITFORENTER** command is that a typed message is included that is to be displayed on the screen. The message can be up to 40 characters in length. The syntax is the same as the **ECHO** command (described below).

Example:
yowaitforenter "<user defined message>

3.5.1.4 Interactive Commands

• **ECHO** types a set of text into a text (dialogue) window.

Example:
echo "<message>
echo "<message>

All text in the line following the ECHO command will be displayed in a window. If there is more than one line of text, simply continue to the next line beginning with the ECHO command again. Only the text will appear in the window, not the word 'echo.'

If more than one window is desired, the **NOP** command can be used between echo commands. This is useful for dividing long segments of text up into different boxes so operators aren't required to do as much scrolling. (See Using the Dialogue Windows, Section 3.3 , for further information).

• **SPEAK** plays a digitized voice file on a SoundBlaster card. The command must be followed by an **ECHO** command to hear the sound.

Example:
speak tada.voc

echo "hello

This feature has only been successfully tested on true SoundBlaster cards. SoundBlaster compatible cards that have been tested did not work.

The sound will only be played if the /sound command line option is used when running 3D World. Additionally, **USEDIGITIZEDVOICE** must be called before these sounds will play.

• **SPEAKONLY** plays a digitized voice file on a SoundBlaster card. This feature differs from **SPEAK** in that an **ECHO** command does not have to follow in order to hear the sound.

Example:
speakonly tada.voc

This feature has only been successfully tested on true SoundBlaster cards. Soundblaster compatible cards that have been tested did not work.

The sound will only be played if the /sound command line option is used when running 3D World. Additionally, **USEDIGITIZEDVOICE** must be called before these sounds will play.

3.5.1.5 Data Collection Commands

• **CLOCK_ON** starts an on screen countdown counter. The start time is specified in the command, and the clock counts down in seconds.

Example:
clock_on 500

• **CLOCK_OFF** stops the on screen countdown counter and erases it from the screen.

Example:
clock_off

• **INPUT** displays an input box on the screen that accepts input from the screen and saves it to the data file. The specified prompt to be displayed cannot exceed 40 characters of text.

Example:
input "<specified prompt>

• **PLAYBACK** plays back a series of keystrokes which were previously recorded as a .mov file.

Example:
playback move.mov

• **NOPLAYBACK** ends the playing back of a series of recorded movements before the set of movements have been completed.

• **POINTTO** Each occurrence of this command will read one line of the 'input file', so it is important that there are the same number of entries in the 'input file' as there are 'pointto' commands. The input file consists of the following:

-Position (x,y,angle)
-Question that the subject sees (256 characters max.)
-Correct heading
-Correct distance

A yellow cross hair will appear in the middle of the screen when the subject is asked to point to an object.

• **RECORD** records operator moves to a specified file having the extension .mov.

Example:
record move.mov

In order to end the move recordings, a waypoint must be positioned at the desired endpoint and contain **NORECORD**. The **END** command will also stop the recording process. The recorded commands include all keys active during a "PLAY" session - currently movement, overhead help, opening doors, and quitting the program without the use of an **END** command. The record command does not record <Enter>. The input for the moves is the same as in the **USEKEYS** mode. The RECORD command is useful for recording operator movements for research purposes. Playing back recorded movements can also be useful, for example, to take tours of an environment without operator interaction.

• **NORECORD** stops recording moves and closes the movement file from the last **RECORD** command.

• **RECORD_TIME** records the current time on the down counter to the data file. It will also include a specified comment if present.

Example:
record_time "<comment>

• **SPIN** allows for movement to spin in a circle. **SPIN** must be coupled with the **CLOCK_ON** command because forward and backward movement is not allowed until the clock is at zero. This command gives the sensation of looking around without moving.

Example:
clock_on 10
spin

• **SWAT** accepts input from the screen. This command is a more specific form of input. SWAT stands for Subjective Workload Assessment Technique and is a research tool used to study operator workload (ref%%%). It displays a prompt on the screen asking the operator to enter Time, Effort and Stress SWAT ratings then saves them to the data file.

Example:
swat

• **TIME** records the elapsed time from the 3D World internal timer.

• **TIMEROFF** disables the 3D World internal timer. This allows for the use of 3rd party software that may interfere with the timer in 3D World.

• **TIMERON** enables the 3D World internal timer.

• **YOCOMPASS** is used in conjunction with the command line argument '/comm'. It is identical to the **YO** command in syntax, but after displaying the message on the screen it sends a 'G' to COMM port 2 and then waits for a 'G' from the external computer before continuing. **WARNING:** ***This command has the ability to lock up the 3D World program if used in conjunction with the '/comm' argument and there is no external computer attached.***

3.5.1.6 Overhead Image Commands

• **SHOWMAP** displays a picture of the overhead picture defined in the 3d.map file. <F10> must be pressed to exit the **SHOWMAP** mode.

• **SHOWPIC** displays a .pcx file centered in the screen. For a better guarantee of positive results, make the picture be a multiple of four in width (i.e. 64 or 68 pixels wide, not 65 pixels). The picture should not be any larger than 320 pixels wide by 200 high. Generally, this should be followed by a **WAIT** or **WAITFORENTER** command to allow the operator time to view the picture.

Example:
showpic map.pcx

• **SHOWPALPIC** is the same as **SHOWPIC**, but also loads the picture's palette. The previous palette is faded out, the picture is loaded, and the new palette is faded back in. This is useful if a palette file does not exist for the current images being used.

Example: showpalpic map.pcx

3.5.2 Egavga.bgi

The graphics driver.

3.5.3 Map.3dm

The actual map of the environment created in the Map Editor

4.0 CREATING AN ENVIRONMENT

In this section, we will be discussing how to actually build the environment using the World Editor and the Map Editor. The World Editor is not required to create an environment. Its purpose is to simplify the development process *especially* when creating the Icon files. If you choose not to use the editor, you must manually edit the Icon files as described in Section 3.2.

4.1 Understanding and Using the World Editor

The World Editor is a development tool which simplifies the creation of an environment by allowing the operator to select options from a menu. You are able to access the Map editor, the DOS editor, a Paint Program, and DOS Shell from within the editor. Perhaps the most important feature of the editor is the ease in which it allows you to create the icon description files (objectdata.def and mapdata.def). The World Editor is not required to create an environment. You can create an environment without ever using the World Editor, however, it can be very helpful in certain ways as we will explain. To access the World Editor, you must have the editor.exe file in your 3d directory, then type 'editor' and it will appear on the screen. You should be able to create a complete working environment by using the selections from the World Editor menu. See Figure 20 for an example of how the World Editor appears on the screen:

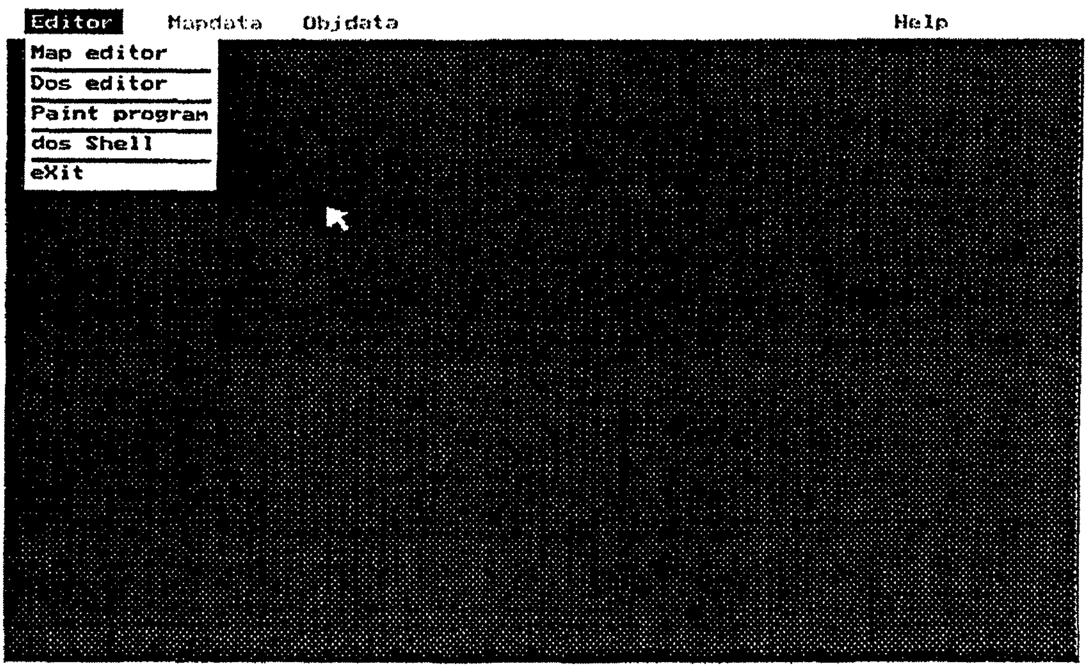


Figure 21. Example of World Editor screen

4.1.1 Editor Menu Selections

Map Editor: Selecting this will display the Map Editor for editing the drawing board.

DOS Editor: This will put you in edit mode in DOS.

Paint Program: This will bring up the Neopaint Paint program for creating/editing .pcx files.

DOS Shell: This puts you in MS DOS mode. You must exit to return to the editor.

4.1.2 Mapdata/Objdata Menu Selections

Selecting these will allow you to create/edit the Mapdata.def and Objectdata.def files and create icons for use in the Map editor. The icons represent the image (.pcx) files that comprise the environment. You will build the environment using these icons on the Map Editor drawing board. Using the World Editor is much simpler than manually entering the field values in the file as described in 3.2. To edit the Mapdata.def file (for example), you would click on mapdata.def, then choose Edit from the menu bar. The following illustration represents what you will see on the screen.

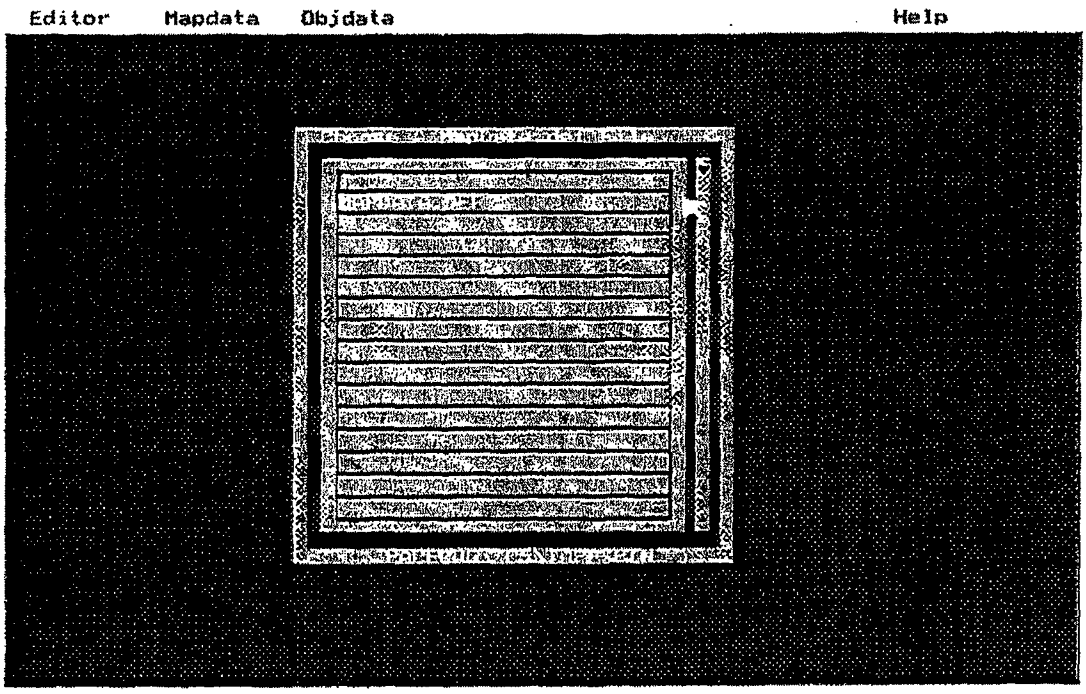


Figure 22. Example of mapdata.def menu

Although the instructions on the box say to choose a file, you should only have one file available. You should select mapdata.def using the RIGHT mouse button to open the file. Figure 22 is an example of how the screen will look once the file is opened.

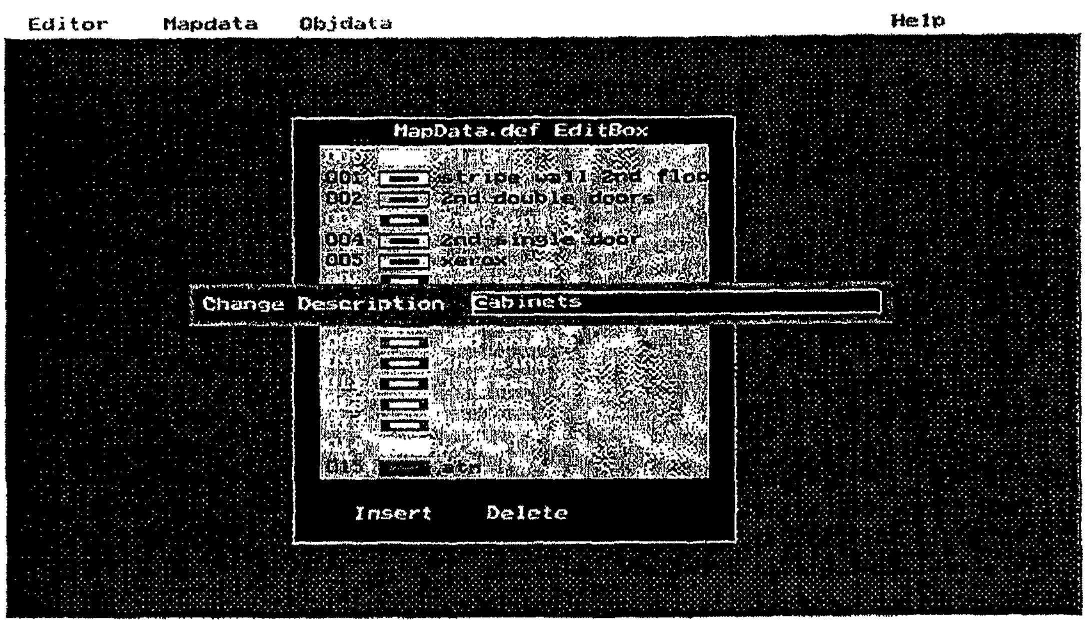


Figure 23. Example of mapdata.def menu

When using the World Editor, the mapdata.def file is organized as described in 3.2.1.: the first field is the item number, the second field is the map piece icon's image, and the third field is the item description. The difference is that the actual icon appears in the second field instead of a field value. You would edit this file to add wall icons, change an icon color, or change an icon's descriptive name. To edit this file, click on a line with the right mouse button. Once you click on a line, that item's description will appear on the screen as shown in Figure 22. You can edit the description by typing in the corresponding block. If you are creating a new icon, the block will be blank, and you should type in the name you choose. Once you have finished, press enter. The only editing you will do is to the second and third fields of the file; icon image (as discussed below) and item description. You will not edit the first field (item numbers) in the World Editor. This is done automatically.

IMPORTANT: *The icons in the mapdata.def and objectdata.def files will be used as map pieces and represent the .pcx image files in the World Database (3d.map) file. Therefore, they should be listed in the SAME order as the .pcx files in the 3d.map file. An image description DOES NOT have to be identical to the .pcx filename since it is for descriptive purposes only, but the .pcx filenames listed in 3d.map MUST be in the 3d directory. Note: You can change the name of a .pcx file while editing the .def files by pressing 'c'.*

Next you will be selecting the foreground and background colors for the item's map icon. The icon will represent the map piece that you will be placing on the drawing board so you may want to try and relate the icon colors to the .pcx image. To choose the colors, press either F for foreground or B for Background. See Figure 23.

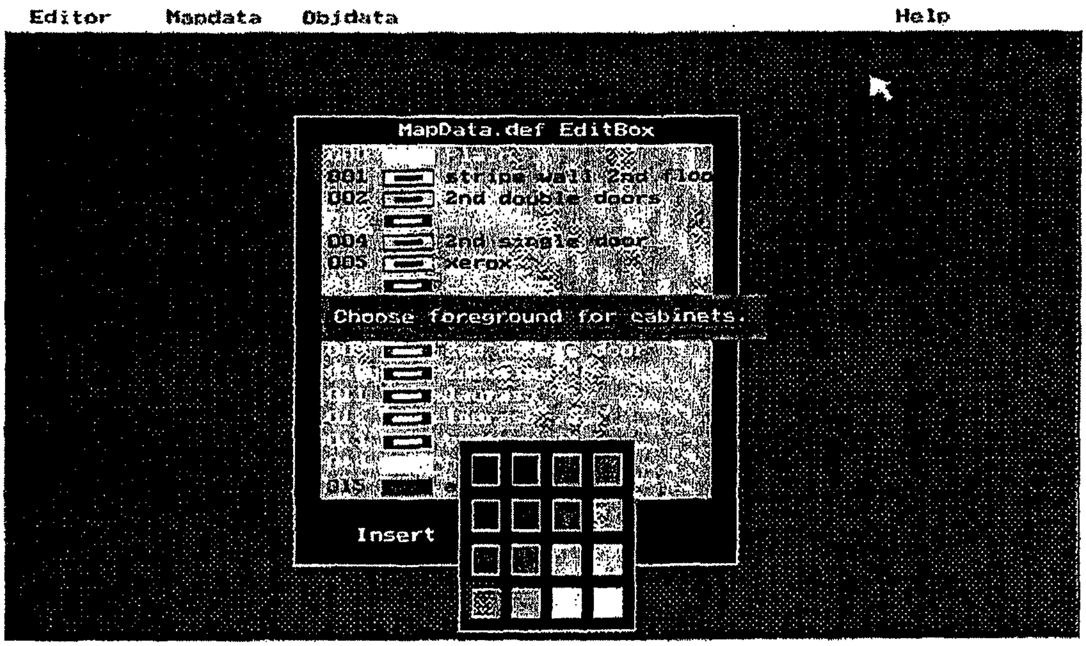


Figure 24. Example of editing the foreground color for the map piece icon in the mapdata.def file

Figure 23 shows how the screen will appear if you're editing the foreground. To choose a foreground color, simply click on a color with the left mouse button. Repeat the process for background color.

Editing the *Objectdata.def* file is identical to editing the Mapdata.def file, with the exception of the shape of the icons. You are permitted to change the shape of an icon in the objectdata.def file. Do this by pressing the 'x' key and choosing any of the patterns on the menu.

IMPORTANT: When you have completed editing a .def file, you must press <Enter> to save the file. ESC will NOT save the file.

4.1.3 INSERT and DELETE

At times, you will find it necessary to insert or delete an item from the .def files. As seen in Figure 22, there are two selections, insert and delete, which you may use to do just that. To insert an item, select the item in the file that you want directly below the inserted item. Click on it with the left mouse button. A line will be inserted in the file which reads "empty." Click on that file with the right mouse button and edit as usual. To delete an item, click on it with the left mouse button.

IMPORTANT: When you add or delete a file from the mapdata.def or objdata.def files, you MUST manually insert or delete the item in the 3d.map file in the same order as it appears in the .def file.

4.2 Understanding the Map editor

The world editor provides a graphical interface called the Map Editor for use in editing the 3D environment. It works similar to many of the standard commercial paint programs on the market using selection lists and multiple drawing tools.

For an example of what the Map Editor looks like, see Figure 25.

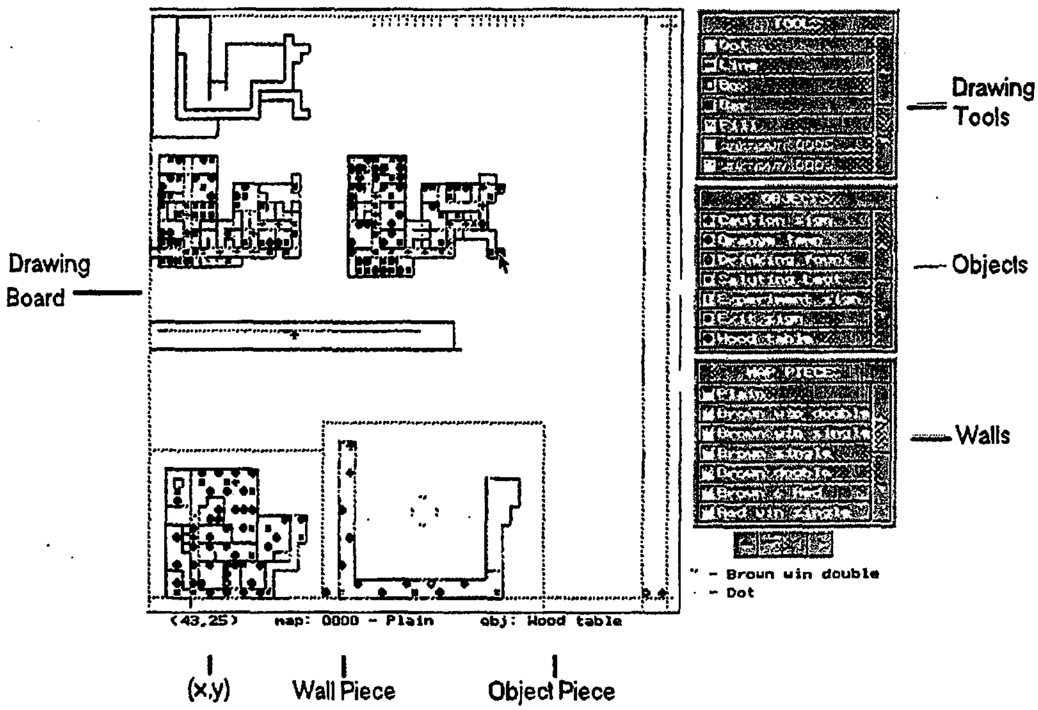


Figure 25. Map Editor

Drawing board: the bird's eye view drawing area representing the environment

Drawing Tools: Lists the tools available for drawing the environment

Objects: Lists the object image files and their icons as defined in the objectdata.def file

Walls: Lists the wall image files and their icons as defined in the mapdata.def file

(x,y): displays the x,y position of the mouse cursor on the drawing board. Note that the upper-left hand corner is (0,0) and both x and y increase toward the lower right corner

Wall Piece: Displays the name of the wall map piece that the cursor is pointed to on the drawing board. If pointed to an object or nothing at all, this will read 0000 - Plain.

Object Piece: Displays the name of the object map piece that the cursor is pointed to on the drawing board. If pointed to a wall or nothing at all, it will say nothing.

4.2.1 Understanding the Map Editor Coordinate System

4.2.1.1 The (X,Y) Plane

• X increases as you move to the right (east); Y increase as you move down (South):

Note that this is a left-handed coordinate system. As a result, when doing math to calculate coordinates, the Y value must often be negated since the standard math model uses up as positive Y.

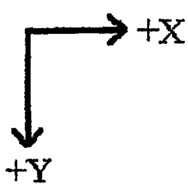


Figure 26. (X,Y) Plane.

4.2.1.2 Angles

• +X is 0; angle increases counter-clockwise, in degrees:

This is conceptually the same as the mathematical model where the angles start from east and increase as the angle goes in the direction of north. Note that when using the /showcoords command line option and **POSITION** in the waypoint command file, this is the system being used.

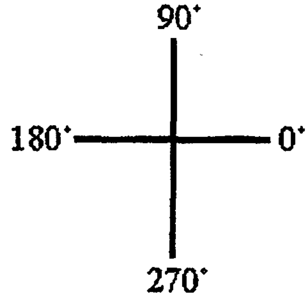


Figure 27. Angles in the Map.

To find the position of a block, always start with the mouse in the upper left-hand corner at (0,0). Move the mouse from this position to the desired position. If you do not start from the upper left-hand corner, you may get an incorrect reading.

4.2.2 Building the Environment Using the Map Editor

The Map editor, as described earlier, is basically a canvas where we draw, or build, an environment. Selection boxes allow the user to easily choose from a large number of drawing tools and images by using a mouse. The selection boxes are located to the right of the drawing board and are categorized into three groups: drawing tools, objects and walls. Below is an example of the Tools selection box.

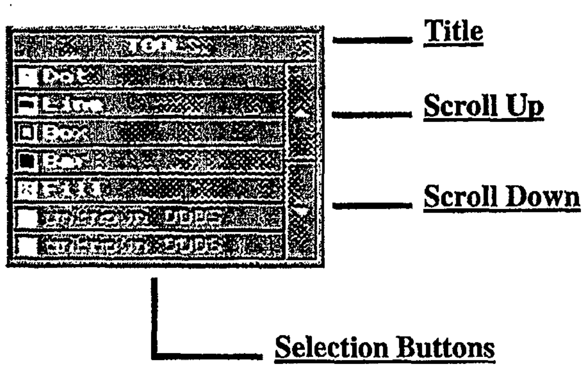


Figure 28. Map Editor Selection Box

The scrollers shift the list of items up or down so that all items can be viewed. You can select an item by simply clicking on it.

4.2.2.1 Tools

Tools affect the way that objects and map pieces are drawn on the screen with the mouse. They are provided to speed up the map drawing process.

The tool selection box is located on the top-right hand corner of the map editor's screen. Tools can be selected by moving the arrow cursor over the item and clicking on the cell of the desired tool.

The tools that are currently available are: dot, line, box, bar, and fill. The **dot** tool is used to place one item on the map at a time. When clicking on the map using the dot tool, one space is filled with either a wall or an object, depending on which type of piece is selected. When dragging the dot tool, a scribbling effect can occur if the mouse is moved too quickly. Drawing will continue as long as the mouse button is held down.

The **line** tool provides for simple line drawing. If the mouse button is clicked, this tool acts like the dot tool, but isn't as precise. The standard use is to drag the mouse from the starting position to the ending position of the line then release the mouse button.
The **box** tool creates a box. This is useful in initially creating rooms by setting up the outside walls. The clicking and dragging functions work similar to the way the line does.
The **bar** tool creates a filled box. This tends to be most useful as a large area eraser. The clicking and dragging functions work similar to the line and bar.

The **fill** tool is very common in professional paint packages. To fill an area, click on the area that should be filled. The fill will not cross wall boundaries. When filling in map pieces, all pieces connected to and of the same type as the original are changed to the currently selected map piece. If you begin a long fill and change your mind, you can terminate the fill by pressing the space bar or most any other key. Sometimes the fill tool will not fill an entire area due to memory constraints. This usually occurs only when filling areas of more than about half of the world. If this happens, simply follow up with additional fills or with the dot or bar tool.

4.2.2.2 Wall Pieces

The third selection box from the top is the Walls selection box. It contains a list of all of the wall pieces that can be used in designing a world. To create rooms or add walls to the map, select the type of tool you want to use from the Tools selection box. Then go to the Walls selection box, and click on the wall piece you want to place in the environment. You should then move the cursor over the map where you would like to place the wall, then press the left mouse button. You will see a map piece appear on the map.

There are two ways to erase a map piece. The first is to select an "Empty" item from the Walls or Objects selection boxes depending on what you want to erase. If you have many wall pieces, it may be toward the end of the list. To avoid scrolling endlessly down, scroll up. The selection boxes allow scrolling in both directions and wrap around when the end or beginning of the list is found. Select and place the "Empty" piece on the map. Doing this will replace, and therefore erase the previous walls.

The second, and far more convenient, way to erase walls is with the right mouse button. Clicking on an item with the right mouse button performs the same as using the "Empty" piece with the left button. The advantage is that you don't have to move your mouse over to the selection bar, scroll up, then go back to where you were, instead, you just erase.

4.2.2.3 Objects

Like the walls, objects have their own selection box. Object pieces function almost identically to the map pieces. Just select an object and begin drawing with it, or use the right button to erase. The main difference between the two when creating the environment is that the walls are placed on the perimeter of a coordinate block, and are thus arranged as horizontal and vertical pieces only. Objects, on the other hand, are always placed in the center of a block. Therefore, using an object at a particular set of coordinates does not preclude the use of north-south and/or east-west wall(s) there. Just remember, only one object may occupy one block, and only one map piece may occupy one side of a block.

Just before the "Empty" object in the selection list, there are 8 arrows. By placing one of these on the map, the starting position of the operator is defined. The arrow specifies the direction the player will be facing when the 3D World viewer loads this world. If more than one of these arrows exist in the world, the one closest to the southeast corner is used. However, it is advised that only one be used to ensure future compatibility.

5.0 VIEWING THE ENVIRONMENT

5.1 Starting the Program

5.1.1 Hardware Requirements

3D World is written in Borland 'C' 3.1 and will run adequately on any IBM PC with at least an Intel 386 processor (or equivalent) with 8 megabytes installed RAM. 3D World is also capable of playing '.mid' and '.voc' files through an authentic Sound Blaster Pro sound card.

5.1.2 Configuring the Computer

3D World uses expanded memory, and the config.sys file must be configured to allow this. In the device line which contains emm386.exe, RAM and h=255 must be added. If the statement NOEMS occurs in this line, it must be removed.

Example:
Device=c:\dos\emm386.exe RAM h=255

5.2 Running 3D World

To begin the program with the default parameter files, simply type 3d<ENTER> at the command prompt. There are several parameters that can be changed when starting 3D World, and the following is a list of the additional available command line options. To use these when running 3D World, type:

3d /<command>

- **/comm**

  This tells 3D World that there is a computer hooked up to COMM port 2 and that whenever a 'yocompass' command is encountered in the waypoint file, the 3D World program will send a 'G' to the computer hooked to COMM port 2 and then will wait until the external computer sends a 'G'. **WARNING:** ***This command has the ability to lock up the 3D World program if used in conjunction with the 'YOCOMPASS' argument and there is no external computer attached.***

- **/data**

  This tells 3D World that subject data is to be collected. The information that is required is the subject's initials and a ten digit identification number. An output file is created based on the information given here. If that file already exists, a prompt will be displayed asking if the file should be overwritten. If the answer is yes, the old file will be deleted and all of its information will be lost.

- **/lowres**

  Using this option causes 3D World to run in a low-resolution mode. This results in significantly better performance on low-end machines. However, this mode is not recommended for machines capable of running at a normally acceptable frame rate because 3D World run in low resolution has severely degraded image quality.

- **/p=*SomeNumber***

  This tells 3D World what size to make the wall and object textures after they are loaded. The default size is 128 by 128 pixels. *SomeNumber* represents this size in pixels. The images must be square, have sides that are a multiple of two in length, and must be less than or equal to 128. So, 2, 4, 8, 16, 32, 64, and 128 are the only valid values. The only reason to use a value other than 128 would be to save EMS memory.

- **/showcoords**

  This has been made for debugging purposes. It displays the operator's position and facing angle while running 3D World. This can be useful for detecting misplaced waypoints.

• **/sound**

By including this option, SoundBlaster support is enabled, therefore, this option requires a SoundBlaster Pro with the following drivers: SBFMDRV, SBMIDI, and SBSIM. If this option is used when a compatible card is not present, 3D World will generally exit with an error message. However, it may lock up the system with some cards, requiring a reboot.

• /w=***SomeNumber***

The default waypoint command file is 1.way. This command accesses other waypoint command files. *SomeNumber* matches the number used in the alternate waypoint file name.

5.3 Navigating in the Environment

Movement in the program is controlled by the number pad and/or the cursor keys. Figure 27 describes the function of the number pad keys:

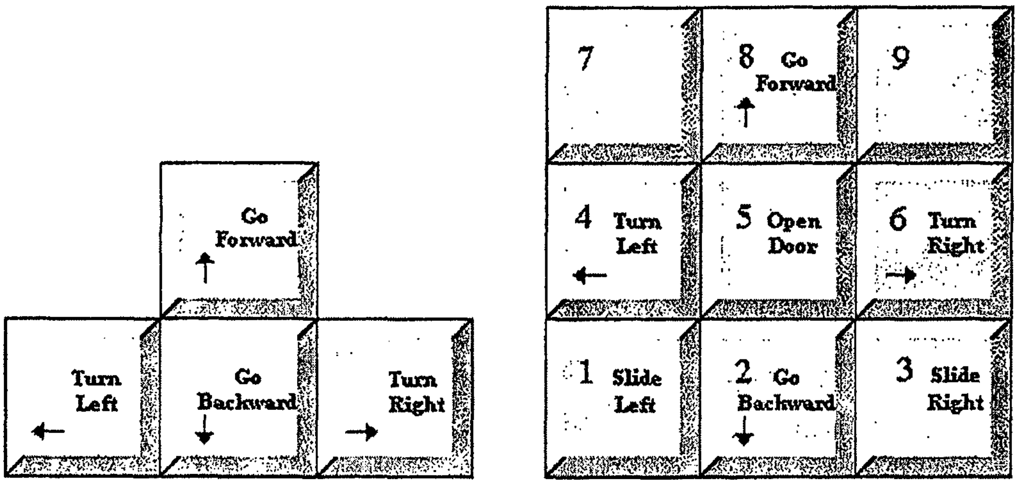


Figure 29. Navigation Keys

There are other keys and key combinations that are active during navigation as well.

SPACEBAR

The space bar is used to open a door. In the simulated environment, the operator must be facing the door (within 45 degrees, non-inclusive) and be closer than 4 feet away.

USEKEYS/SHIFT

In the **USEKEYS** mode, holding down <SHIFT> while walking causes the operator to move twice the normal speed.

ALT-H
Displays the HELP Screen

ALT-M
Displays the Overhead Pictue

ALT-Q
Quits the program and closes the output file.
While navigating through the environment, waypoints will be activated. Navigation control will shift to the waypoint commands then return to the navigation keys when all commands in the waypoint have been executed.

5.4 Using Dialogue Windows

Dialogue windows are used to present the operator with textual information. A dialogue window simply displays text on the screen and can be scrolled up and down with the arrow keys. When the operator has completed reading the text in the window, he or she can close it by pressing <ENTER>.

A Dialogue Window is created whenever the ECHO command is used (see Section 3.5.1.4). There is no limit to the amount of text you can put into a dialogue window, but if you have a lot of text, it is suggested that you divide it up into several windows. You would do this by using the NOP command between echo commands in the waypoint file.

Example:

echo "Today is your first day as an Office Assistant in the Psychology Department. You
echo "will be running errands, making copies, purchasing materials from the bookstore,
echo "and performing other miscellaneous tasks.
NOP
echo "You should see your supervisor to get a list of the tasks you are to perform.

In the above example, two dialogue windows will be displayed. The first one will contain all the text that follows the ECHO commands in the first three lines. The operator will then close that dialogue window and then the next window will appear on the screen. This screen will contain all the text following the next ECHO command. Figure 28 is an example of a dialogue box as it appears on the screen.

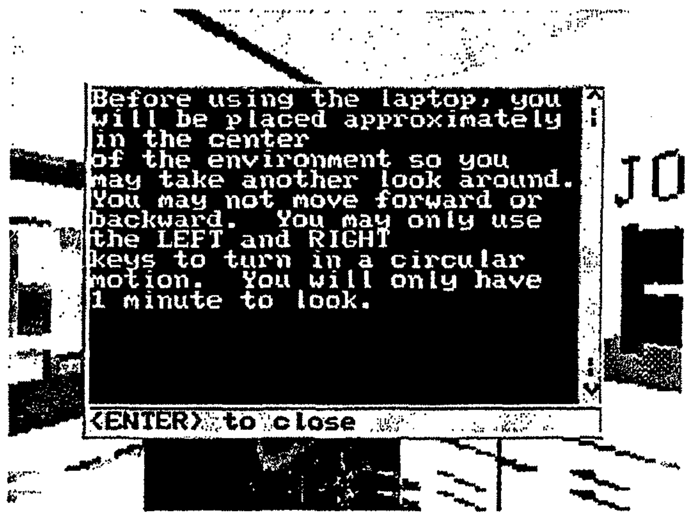


Figure 30. Example of Dialogue Window

6.0 DATA COLLECTION / OUTPUT FILES

6.1 .out file

3D World provides for automated data collection. Type 3d/data to start the 3D program. When 3D World begins, it will ask for the first and last initials, and identification number of the operator. This information is used to construct the name of the .OUT file as well as to give a short header for each line in the file. The following is the construction of the filename:

first and last initials + identification number + .out

Example:
ss0956.out

The .out file will be saved in the same directory as the program.

Each line in the .out file will start with the first and last initials and identification number. The next part of the .out file will be a description of what waypoint command was used. The final part will be the input. The line is divided by colons, so it is easy to import the file into a spreadsheet. The following is what each command will give as an output:

Command	First and Last Initials	ID Number	Message	Type of Input	Type of Input
input	Initials:	ID#:	Message given to operator:	Input from operator	
record_time	Initials:	ID#:	Message specified in command:	Time from counter	
swat	Initials:	ID#:	SWAT:	Input from operator	
time	Initials:	ID#:	Elapsed time:	Time	
copier*	Initials:	ID#:	Copier:	Number	
file_on*	Initials:	ID#:	File:	Name	
phone*	Initials:	ID#:	Phone:	Correct or Incorrect	
menu*	Initials:	ID#:	Menu:	Quantity: (default=1)	Food Item

*The copier, file_on, phone, and menu commands are described in Appendix A.

If none of these commands appear in a waypoint file, or if they were not activated, the .out file will be blank.

6.2 .mov file

All files ending in .mov are generated by commands in the waypoint command file. They define a series of moves and consist of a recording of the character codes written to the keyboard buffer. The recorded commands include all keys active during a "PLAY" session. In order to create a .mov file, you will need to run the program with the **RECORD** and **NORECORD** commands in the waypoint file. 3D World will record all movements between those two commands, beginning immediately following **RECORD** and ending with **NORECORD**. You would then remove those commands from the waypoint file, and insert the **PLAYBACK** command to play back the recorded movements. See Data Collection Commands under the Waypoint Running File, Section 2.1.3 for more information on these commands.

7.0 QUICK-STEP GUIDE TO CREATE A VIRTUAL ENVIRONMENT

Here are six easy steps to begin building an environment in 3D World. It is assumed that wall and object .pcx files have already been created or taken from a library. All files must be in the same directory to run the program. The 1.way file and the 3d.map file must be created before the program can run.

Step 1: Create the World Database (3d.map) file.
Step 2: Edit the mapdata.def and objdata.def files.
Step 3: Draw the environment.
Step 4: Create a waypoint (1.way) file
Step 5: Test the environment.
Step 6: Finish the waypoint file.

Files that are necessary to begin building an environment:

1.way	3d.exe
3d.map	editor.exe
objdata.def	editmap.exe
mapdata.def	initmap.exe
tools.def	egavga.bgi

Step 1:
The .map file must be created to set the parameters for the environment and to list the .pcx files that are to be used as walls and objects.

- Wall pieces should be listed under the [pic] heading.
- Object pieces should be listed under the [obj] heading.

The following is an example including how to add an open door, a wall, a closed door, and a table.

```
[map]
map.3dm

[parameters]
eyelevel 5.0
stepheight .50
speed 31
stepdist 3
turnrate 100
```

[pic]
thrudoor.pcx opendoor window
wall.pcx
bluedoor.pcx door

[obj]
blutable.pcx

[overhead]

[help]
help.pcx

Step 2:
There are two different methods of editing the mapdata.def and objdata.def files. It can be done manually or the editor can be used. The editor is the easier method; however, there are times when you may want to edit the files manually.

To use the editor:

• Type editor at the prompt.
• Left mouse click on the MAPDATA pull-down menu.
• Left mouse click on EDIT.
• Right mouse click on MAPDATA.DEF.
• Press <Enter> or right mouse click to change the description. Although it is referring to that specific file, the description does not have to be identical to the filename of the .pcx file.
• Press <Enter>.
• To change the color of the foreground, press <f>.
• To change the color of the background, press <b>. It is necessary to give each piece it's own color combination so the pieces can be differentiated when building the environment. Note: Changing the foreground and background colors do not affect the colors of the actual picture.
• Enter a description for each of the pictures found in the [pic] section of the .map file. Remember to keep them in the same order.
• When all pieces have been described, press q to save the file.
• Repeat this procedure for the objects, starting with the OBJDATA pull-down menu.
• To exit the editor, click on EDITOR.
• Click on exit.

Step 3:
Create a new map using initmap.exe. When this command is used, map.3dm will be created to give a new blank map.

- Type initmap at the prompt.

After map.3dm is created, it should be edited to place the walls and objects into the new environment. To edit the map, use editmap.exe.

- Type editmap at the prompt.
- Place walls and objects in the environment.
- Place a starting arrow in the environment.
- To quit from editmap.exe, type q, then y to save.

Step 4.
A rudimentary way point file must be created. This file will be modified and expanded later in development of the environment. For the purposes of beginning to build an environment the 1.way file must load the .map file. An example of this is as follows:

```
[start]
showpalpic keyboard.pcx
load 3d.map
play
```

The **SHOWPALPIC** command loads the picture to be displayed while the 3d.map file is being loaded. Note:* You can change the name of the 3d.map file so you can have multiple files in one directory. Simply load the one you want to use.

Step 5.
Test the environment.

- Type 3d at the DOS prompt.
- To open doors, press the space bar.
- To exit from the environment, press Alt-q.

Step 6.
Finish the waypoint command file.

8.0 TROUBLESHOOTING

• Memory is sometimes lost while running 3D World. Here is an example of an error message that may appear when 3D World is exited:

HEAP CORRUPT in DoneMapLocs!

With luck, the computer will run 3D World again, however, normally the system must be rebooted to recover the memory.

• If the palettes do not match on all of the pictures, the picture with the different palette may change the appearance of colors on the other pictures. Therefore, make sure that every picture uses the same palette.

• The .way, .map, and .def files can be edited from within Windows (using Notepad or another editor) or they can be edited in DOS by typing edit at the prompt. Editor.exe, editmap.exe, and 3d.exe must be run from the DOS prompt. 3d.exe usually does not work if Windows has already been started.

• Voice files should be saved in SoundBlaster format (11 mhz, Mono, 8-bit) as a .voc file.

• Make sure any given room has walls completely encircling it. Otherwise, you will be able to walk outside of the environment and the computer will probably lock up.

• If the editor is started from the MS-DOS prompt in Windows, the mouse may not be active. Exiting Windows and starting editor should solve this problem, however, the whole system may need to be rebooted to DOS.

• The editor will allow you to run other programs. However, it has been found that sometimes the other programs do not run correctly from editor. If this is happening, it is best to exit completely from editor and start the new program from the DOS prompt.

• There are times when the editor leaves blocks of color on the screen. This will occur when a foreground or background color is chosen. These can be ignored, they will not harm anything in the environment.

9.0 REFERENCES

Colle, H. A., & Reid, G. B. (1998). The Room Effect: Metric Spatial Knowledge of Local and Separated Regions. *Presence*, 7(2), 116-128.

Colle, H. A., & Reid, G. B. (1999). The Room Effect: Exploring Paths and Rooms with Objects Grouped Categorically and Spatially. Submitted to Ecological Psychology.

Endsley, M. R. (1993). A Survey of Situation Awareness Requirements in Air-to-Air Combat Fighters. *The International Journal of Aviation Psychology*, 3(2), 157-168.

Nygren, T. E., Schnipke, S. K., & Reid, G. B. (1998). The Effects of Workload on Groups formed by Subjective Ratings of the Subjective Workload Assessment Technique. *Proceedings of the 42nd Annual Meeting of Human Factors and Ergonomics Society*, 816-820.

Mailing Address:

Annette McCoy
Sytronics, Inc.
WPAFB, Area B, Bldg 197
Dayton, OH 45433
Annette.McCoy@wpafb.af.mil
937-255-5111

APPENDIX A. - ADDITIONAL WAYPOINT COMMANDS

Included in 3D World, there are additional waypoint commands that were used for a specific experiment. Each of these is designed to work with a particular .pcx file and for a specific purpose. If additional waypoint commands are desired, they must be programmed in the source code.

• **COPIER** accepts input from the screen via the mouse. The command works in conjunction with the copier.pcx file. The command format is: copier "<pcx file name>. See Figure 29.

Example:
copier "copier.pcx

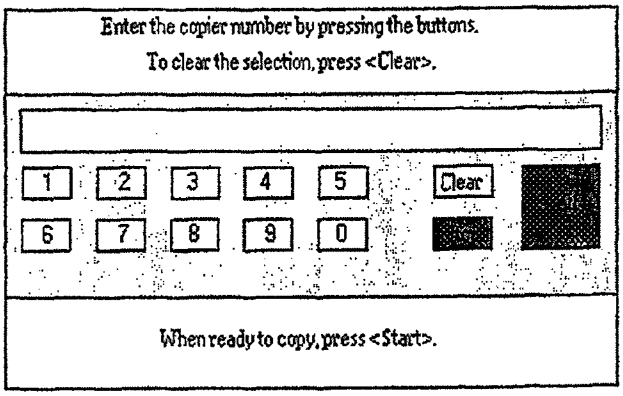


Figure 31. Copier.pcx

When a number is clicked with the mouse it appears in the copier window. When 'Clear' is clicked, the numbers are cleared. When 'Start' is clicked, the input is saved in the data file and the next command in the waypoint file is executed. To replicate this .pcx file with this function, the .pcx file must be 320x200 pixels large. The top left and bottom right corners for each of the buttons should be placed in the following configuration (in pixels):

Button	Top Left	Bottom Right
1	(9,84)	(34,99)
2	(48,84)	(73,99)
3	(87,84)	(112,99)
4	(126,84)	(151,99)
5	(165,84)	(190,99)
6	(9,110)	(34,126)
7	(48,110)	(73,126)
8	(87,110)	(112,126)
9	(126,110)	(151,126)
0	(165,110)	(190,126)
Clear	(225,84)	(255,99)
Start	(271,84)	(310,126)

Figure 32. Copier.pcx Replicate Configuration

• **FILE_ON** accepts input from the screen via the mouse. This command waits until the operator presses a specified key or they leave the current waypoint. If the correct key is pressed, an operator specified .pcx file is displayed, and the subject uses the mouse to make the selection. The command format is: file_on <user specified key>, <.pcx file to display>, <name1>, <name2>, <name3>, <name4>, <name5>, <name6>, <name7>, <name8>, <name9>.

Example:
file_on r, r_file.pcx, Karen Randall, Rowan Regal, Denny Renner, Jeff Richards, Tracy Rogers, Darrell Rolex, Sean Row, Thomas Russo, Daniel Rutski

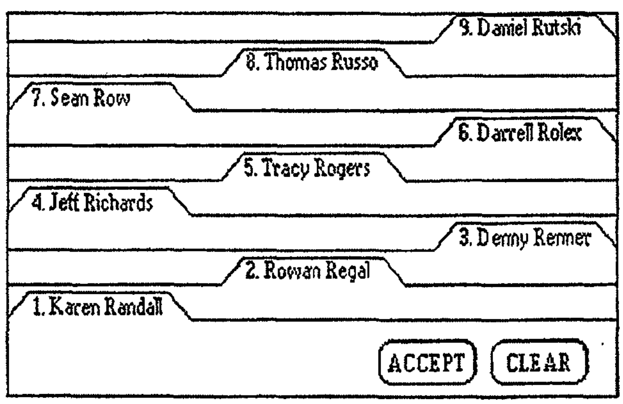


Figure 33. R_file.pcx

When a name is clicked, the output that is specified in the waypoint command appears in the bottom left hand corner. The 'Clear' button clears the name. The 'Accept' button saves the input in the data file and the next command in the waypoint file is executed. To replicate this .pcx file with this function, the .pcx file must be 320x200 pixels large. The top left and bottom right corners for each of the buttons should be placed in the following configuration (in pixels):

File/Button	Top Left	Bottom Right
1	(11,145)	(85,159)
2	(123,127)	(197,141)
3	(235,109)	(309,123)
4	(11,91)	(85,105)
5	(123,73)	(197,88)
6	(235,56)	(309,70)
7	(11,37)	(85,51)
8	(123,19)	(197,33)
9	(235,1)	(309,15)
Clear	(252,171)	(303,192)
Accept	(193,171)	(244,192)

Figure 34. R_file.pcx Replicate Configuration

• **PHONE** accepts input from the screen via the mouse. The command works in conjunction with the phone.pcx file. The command format is: phone "pcx file, <phone number 1>, <phone number 2>, <phone number 3>, <phone number 4>, <voice file for phone number 1>, <voice file for phone number 2>, <voice file for phone number 3>, <voice file for phone number 4>, <voice file for wrong number>. If one of the correct phone number is dialed, the corresponding .voc file will be heard. If a wrong number is dialed, the wrong number .voc file will be heard. When either all four correct numbers are dialed or the wrong number is dialed twice, the command is terminated and the next command is executed in the waypoint file.

Example:
phone "phone.pcx, 2911234, 2914567, 2917890, 2918765, message1.voc, message2.voc, message3.voc, message4.voc, wrong.voc

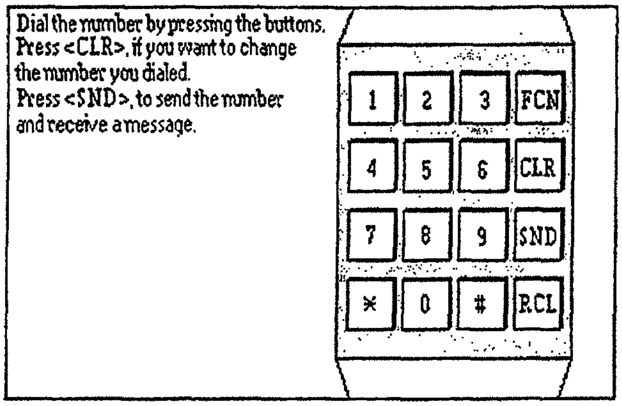


Figure 35. Phone.pcx

When a number is clicked, it appears in the bottom left hand corner. When the 'Clr' is clicked, the numbers are cleared. When 'Snd' is clicked, the input is saved in the data file and the number is cleared. The next command in the waypoint file will be executed when either all four correct numbers are dialed or two wrong numbers are dialed. To replicate this .pcx file with this function, the .pcx file must be 320x200 pixels large. The top left and bottom right corners for each of the buttons should be placed in the following configuration (in pixels):

Button	Top Left	Bottom Right
1	(181,35)	(204,59)
2	(209,35)	(233,59)
3	(238,35)	(262,59)
4	(181,69)	(204,93)
5	(209,69)	(233,93)
6	(238,69)	(262,93)
7	(181,103)	(204,128)
8	(209,103)	(233,128)
9	(238,103)	(262,128)
0	(209,139)	(233,163)
Clr	(267,69)	(291,93)
Snd	(267,139)	(291,163)

Figure 36. Phone.pcx Replicate Configuration

• **MENU** accepts input from the screen via the mouse. The command works in conjunction with the menu.pcx file. The command format is: menu "<pcx file name>.

Example:
menu "menu.pcx

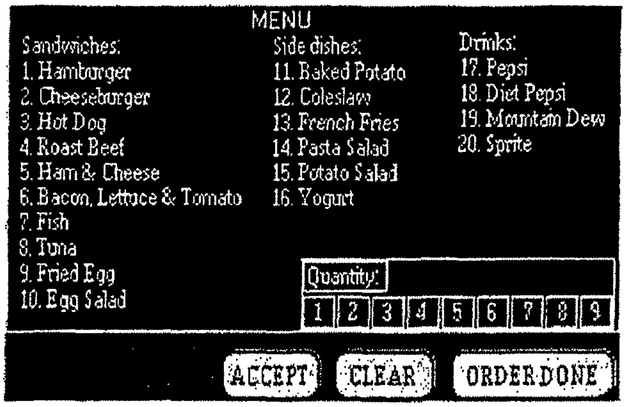


Figure 37. Menu.pcx

When a number is clicked on, the output appears next to the word 'Quantity'. When a food item is clicked on, the output appears in the bottom left hand corner. These particular food items are programmed in the source code. The 'Clear' button clears the order. The 'Accept' button saves the output of the quantity and food item in the data file. The 'Order Done' button exits this command, and the next command in the waypoint file is executed. To replicate this .pcx file with this function, the .pcx file must be 320x200 pixels large. The top left and bottom right corners for each of the buttons should be placed in the following configuration (in pixels):

Quantity/ Button	Top Left	Bottom Right
1	(155,149)	(171,164)
2	(173,149)	(189,164)
3	(191,149)	(207,164)
4	(209,149)	(225,164)
5	(227,149)	(243,164)
6	(245,149)	(261,164)
7	(263,149)	(279,164)
8	(281,149)	(297,164)
9	(299,149)	(315,164)
Accept	(113,176)	(162,198)
Clear	(172,176)	(222,198)
Order Done	(232,176)	(316,198)

Figure 38. Menu.pcx Replicate Configuration

Food Item Number	Top Left	Bottom Right
1	(5,28)	(67,38)
2	(5,42)	(75,52)
3	(5,55)	(53,65)
4	(5,67)	(62,77)
5	(5,81)	(80,91)
6	(5,93)	(123,103)
7	(5,107)	(33,117)
8	(5,120)	(37,130)
9	(5,132)	(58,142)
10	(5,146)	(63,156)
11	(137,28)	(208,38)
12	(137,42)	(189,52)
13	(137,55)	(205,65)
14	(137,67)	(200,77)
15	(137,81)	(204,91)
16	(137,93)	(182,103)
17	(234,26)	(270,36)
18	(234,40)	(290,50)
19	(234,53)	(274,63)
20	(234,65)	(274,75)

Figure 39. Menu.pcx Replicate Configuration (continued)

THIS PAGE INTENTIONALLY LEFT BLANK

DOCUMENT 4

Force/Tactile Feedback System for Virtual Reality Environments

AD-A342328

April 1998

Computer Graphics Systems Development Corp.
Mountain View, CA

THIS PAGE INTENTIONALLY LEFT BLANK

Final Report

FORCE/TACTILE FEEDBACK SYSTEM FOR VIRTUAL REALITY ENVIRONMENTS

DISTRIBUTION STATEMENT A

Approved for public release; Distribution Unlimited

April 3, 1998

CGSD
CORPORATION

Computer Graphics Systems Development Corporation

tel: 650-903-4920
fax: 650-967-5252

2483 Old Middlefield Way #140
Mountain View, CA 94043-2330
415-903-4920 Fax: 415-967-5252
http://www.cgsd.com

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE 4/3/98

3. REPORT TYPE AND DATES COVERED Final Report 4/3/98

4. TITLE AND SUBTITLE
Force/Tactile Feedback Systems for Virtual Environments

5. FUNDING NUMBERS
SBIR
Contract No.
M67004-95-C-0013

6. AUTHOR(S)
Computer Graphics Systems Development Corporation (CGSD Corp.)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Computer Graphics Systems Development Corporation
2483 Old Middlefield Way, Suite 140
Mountain View, CA 94043-2330

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
US Army
STRICOM
12350 Research Parkway
Orlando, FL 32826-3276

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
TOPIT™ User's Guide

12a. DISTRIBUTION/AVAILABILITY STATEMENT
nonclassified unlimited distribution

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

TOPIT™ - Touch Objects Positioned In Time, is a vitural reality system that allows a user, wearing a head mounted display (HMD) to actuate control knobs or dials on a cockpit instrument panel appearing in user's HMD. The actual control knobs/dials are delivered to the user by an x-y-z robot. The instrument panel can be reconfigured entirely in software. A tracker and data glove continually provide the position of user's hand and finger to a computer, and the computer commands the servomechanism system to place the correct type of control in the correct position to be actuated. The servo system has a "touch panel" that contains examples of a dozen or so different types of controls, such as toggle switches, knobs, and push buttons, that are used repeatedly to represent any number of instrument panel controls. One key aspect of the system is building a servo system that moves fast enough to put the knobs/dials in place at the time when the user reaches for it. Another key aspect is achieving precise low-latency tracking of both user's head and user's hand.

14. SUBJECT TERMS
virtual reality, helmet mounted display, robotics, tracker, data glove, cockpit instrument panel, servomechanism, servo

15. NUMBER OF PAGES 44

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
unclassified	unclassified	unclassified	unlimited

CGSD CORPORATION

Computer Graphics Systems Development Corporation

2483 Old Middlefield Way #140
Mountain View, CA 94043-2330
415-903-4920
fax 415-967-5252

Prepared for:

U.S. Army Simulation, Training, and Instrumentation Command
Contract No. M67004-95-C-0013
SBIR A93-316 II – Force/Tactile Feedback System for Virtual Reality Environments

Prepared by:

Computer Graphics Systems Development Corporation (CGSD Corp.)
2483 Old Middlefield Way, Suite 140
Mountain View, CA 94043
Tel: 650-903-4920 Fax: 650-967-5252
http://www.cgsd.com

Table of Contents

List of Figures

List of Tables

1. Introduction

The top-level objective of this SBIR Phase II project was to build a prototype virtual cockpit that included force and tactile feedback. We achieved this top-level objective and all key technical objectives discussed in section 2 as well. We discuss more of each of the technical objectives and our approach in detail later in the report when we present our system concept in Section 3, and Phase II results in Section 4.

System overview and project accomplishment

The user wears a head mounted display that presents stereo imagery of a cockpit interior, including the instrument panel, as well as the out-the-window scenery. A representation of the user's hand is also rendered in the scene. The user may actuate a variety of controls on the instrument panel, and can accurately feel the forces and surface textures of the controls. The simulator can be reconfigured entirely in software to represent different cockpits. The feel of the instrument panel controls is provided by a servomechanism device that places actual physical controls in their correct positions, orientations, and configurations. A tracker and data glove continually provide the position of the user's hand and fingers to a computer. The computer senses the position as the person reaches for a control. Using the extrapolated data, the computer commands the servomechanism system to place the correct type of control in the correct position to be actuated. The servo system has a "touch panel" that contains examples of a dozen or so different types of controls, such as toggle switches, knobs, and push buttons, that are used repeatedly to represent any number of instrument panel controls.

The system is called a TOPIT™ - Touched Objects Positioned In Time. One key aspect of the system is building a servo system that moves fast enough to always have the control in place before the user's hand reaches it. Another key aspect is achieving precise low-latency tracking of both the user's head and the user's hand. The tracking must be accomplished in the presence of the moving metal elements and the electric motors of the servo system; a hybrid magnetic/inertial tracker was developed to meet these requirements. The system has three computers: an SGI Onyx/RealityEngine2 that does the imagery, a Pentium-based PC that does the tracking, and a VME-based servo control system.

The TOPIT Force/Tactile Feedback System concept drawing [Figure 1-1] shows the proof-of-concept demonstrator being used to simulate an aircraft cockpit. The central issue of the feasibility of the scheme is establishing and meeting the timing requirements for determining the touched-object and moving it into place in time.

However, while basic feasibility was established in Phase I, construction of a demonstrator during the Phase II effort required the careful design and integration of mechanical, electro-mechanical, and computer controlled devices to meet project objectives.

Overall, the major technical challenges were met. In particular, robotic hardware was built to position the controls with the speed and accuracy required, and a sophisticated tracker and an alternative tracker were built to provide the accuracies required for position and extrapolation. The most difficult aspect of the program turned out to be getting all of the bugs out of the complex system under severe budget constraints. In this last respect we were largely successful, but not entirely. The main limitations of the final prototype lie in the fine points of getting the software to run completely smoothly and reliability. We view none of the present limitation as being fundamental.

Report organization

Section 1 presents an overview of the project and snapshots of subsystems and components the prototype developed. Section 2 discusses the technical objectives of the project. Section 3 discusses the system concept and implementation, and section 4 compares the results of the phase II effort to the objectives and the original designs for the project. Section 5 presents the conclusions.

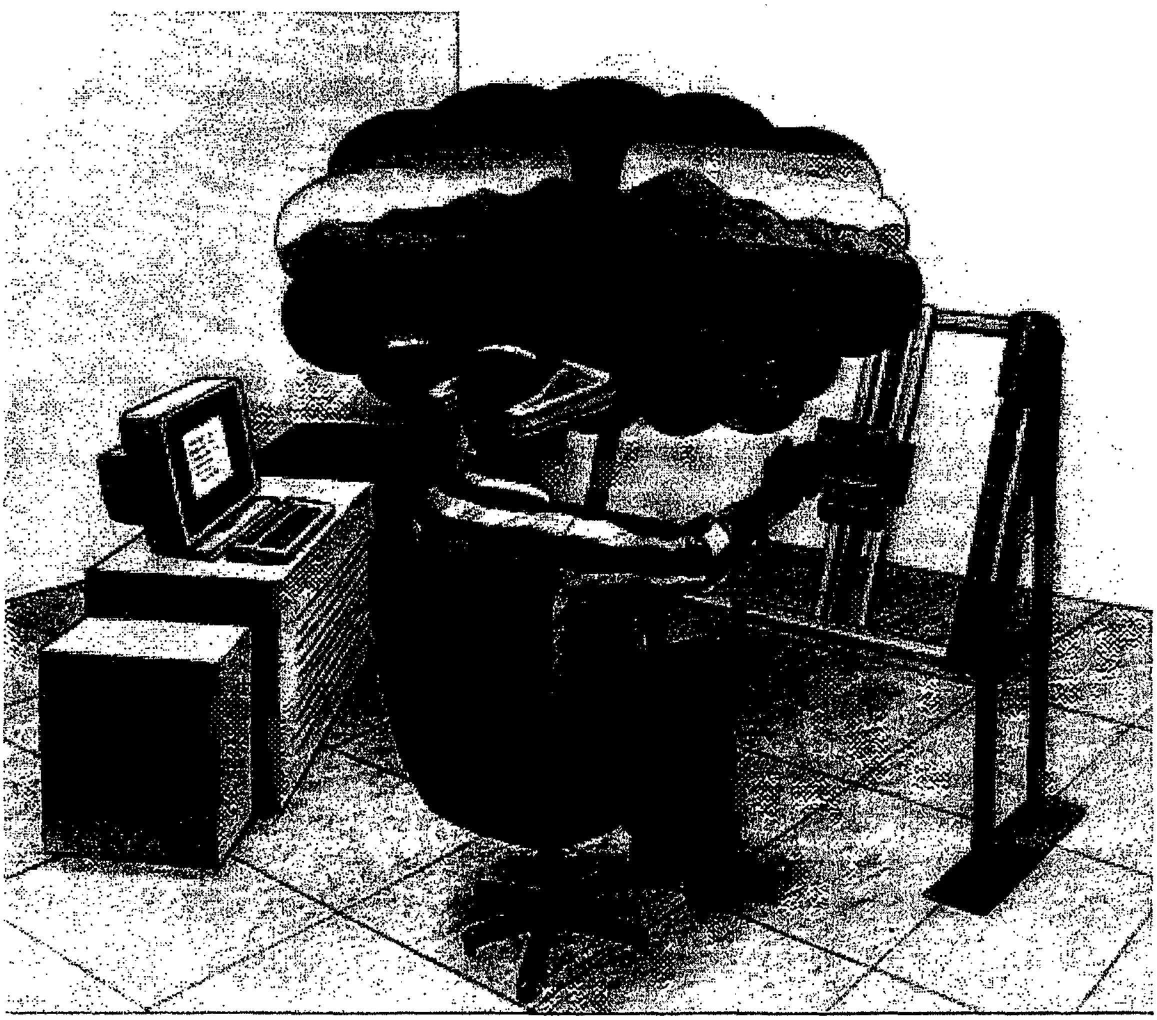


Figure 1-1 TOPIT concept. Physical switches and knobs are positioned in a virtual environment under software control to provide flexible force and tactile feedback.

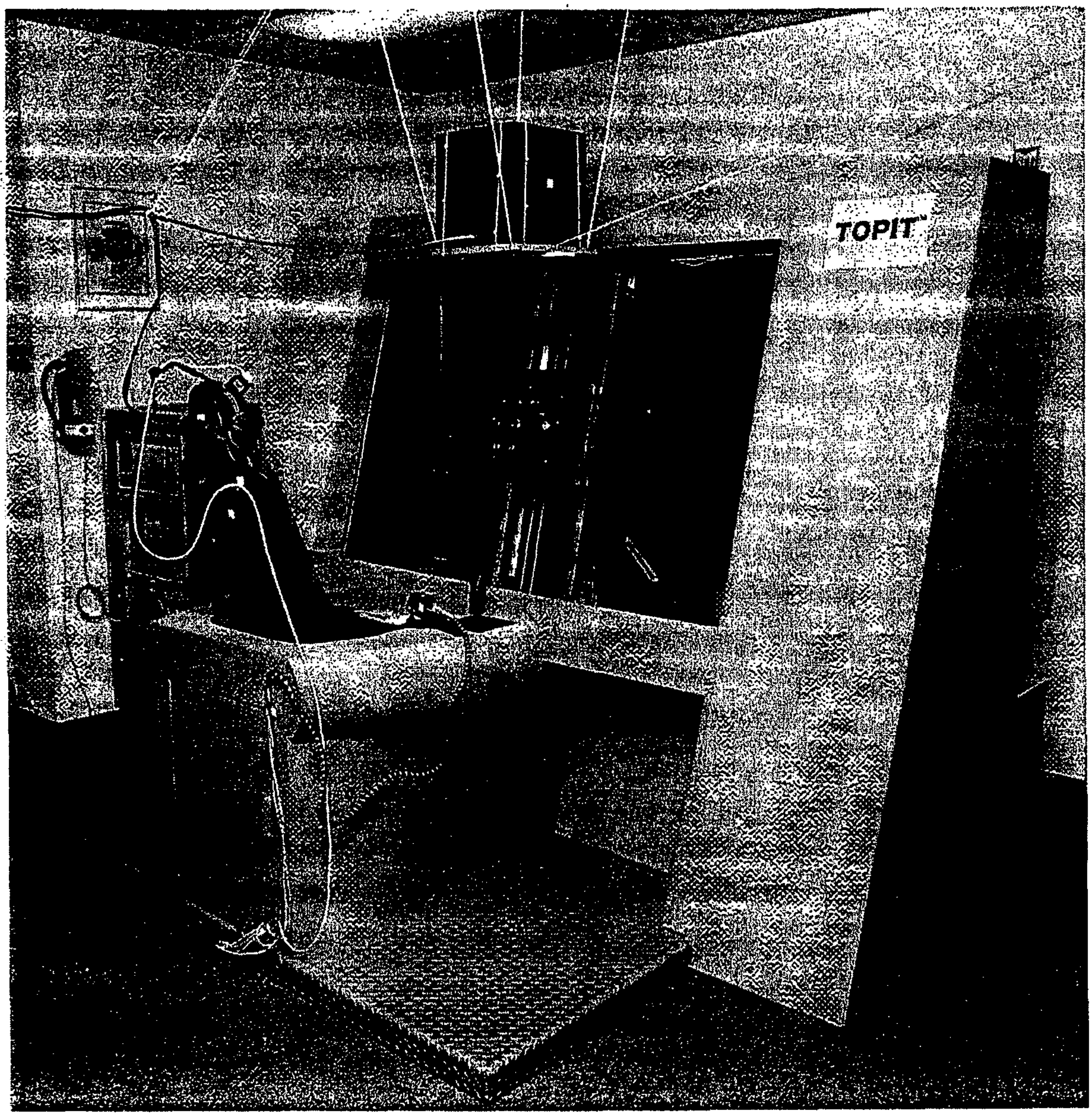


Figure 1-2 TOPIT Prototype.

Figure 1-3.1 User station showing joystick, throttle, instrumented glove, and helmet-mounted display.

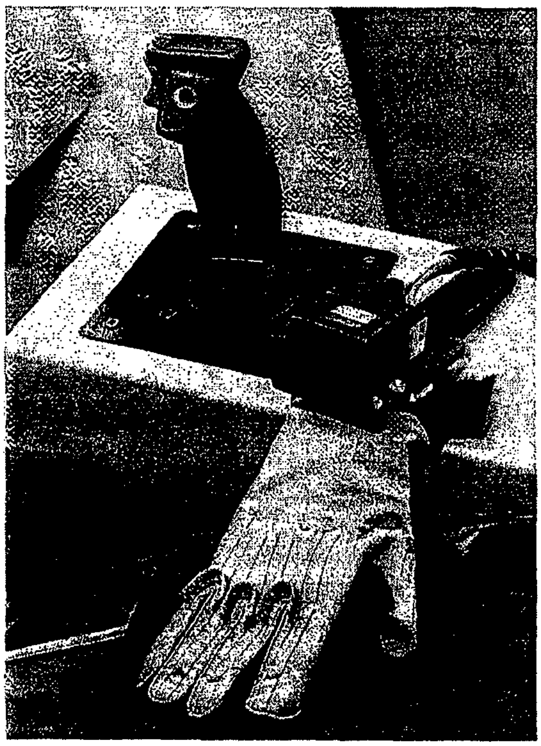


Figure 1-3.2 Joystick and instrumented glove.

Figure 1-3.3 Multisensor.

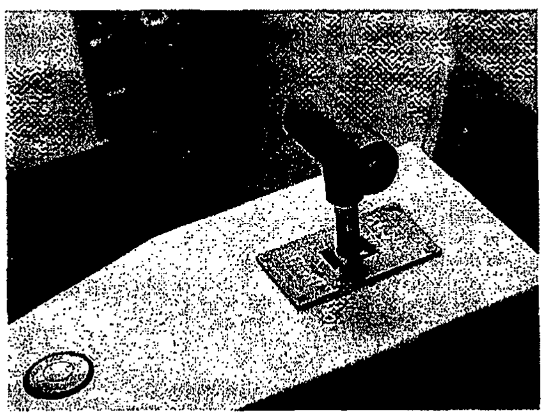


Figure 1-3.4 Throttle and emergency stop button.

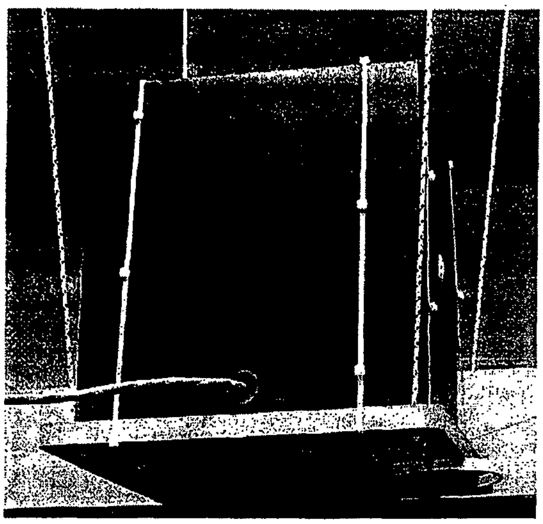


Figure 1-3.5 Magnetic tracker transmitter.

Figure 1-3.6 Y-axis servo motor.

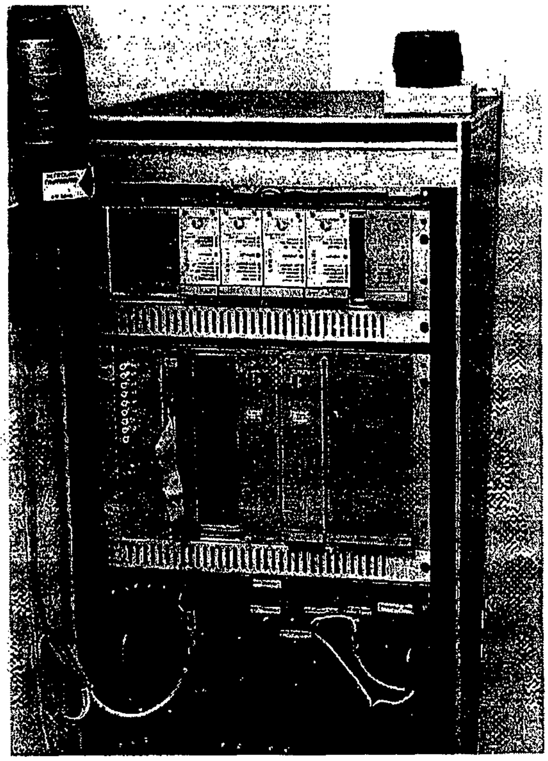


Figure 1-3.7 Servo electronics cabinet.

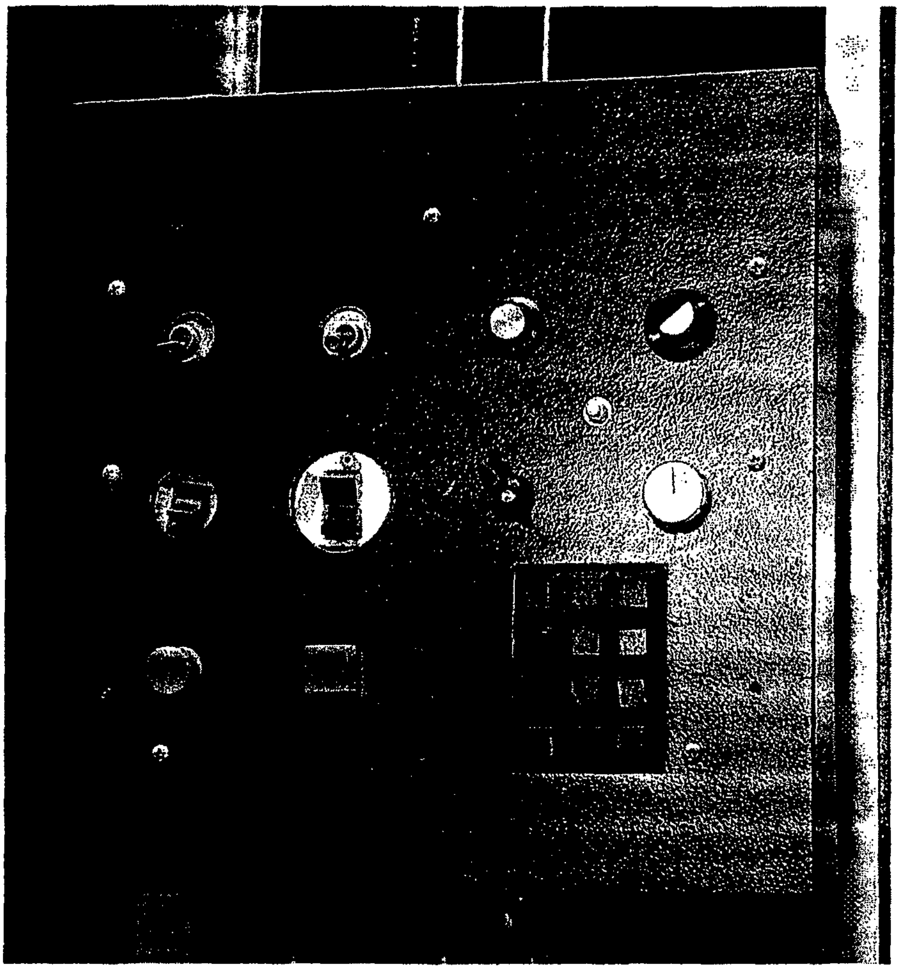


Figure 1-3.8 Touchpanel.

2. Summary of Technical Objectives and Approach

The primary objective of the Phase II effort was to design, construct, and evaluate the TOPIT force and tactile feedback system through a complete implementation of a virtual cockpit. We considered developing a partial implementation, without the visual simulation of the virtual environment. The visual environment, however, was necessary to guide the user to each specific point in virtual space where a virtual control was located. Without the visual simulation, the touchboard could only be guided to mirror hand and finger position, and the demonstration would miss the whole aspect of predicting hand trajectory, selecting the correct control, and fixing the control position in time to be touched. Also missing would have been the aspect of treating head tracker and image generator delays. With so much missing, we concluded that a partial implementation would be unconvincing in proving the TOPIT concept. The approach we adopted paid special attention to the risk areas identified in the Phase I study. The risk areas, identified in the Phase II proposal, and our approach to each key risk area were as follows:

(1) We wanted to build a positioning system that **moved fast enough,** but without excessive size, power, or cost was to be approached through a combination of rapid prototyping, in which the linear transport mechanism for the x-axis positioning was built experimentally using stepper motor and servomechanism implementations, and payload weight was minimized through careful design that encompassed the use of lightweight materials.

(2) We needed to ensure the tracking system provided **sufficient accuracy** in the presence of the electromagnetic noise and moving metal objects of the positioning system was approached by use of a pulsed rather than continuous wave tracker, synchronization of tracker pulses between motor steps, noise minimization by shielding, and by careful tracker transmitter placement. If problems persisted, a noise immune, but somewhat encumbering, mechanical tracker was to be used to support development.

(3) We needed to design hand motion prediction algorithms that predicted which control would be touched while sufficient time remained to put it in place was first approached at the system level using the basic hand motion data obtained in Phase I. These data bound the performance of the algorithm. However, considerable experimentation were made to fine tune the algorithms. Also, an alternative tracking system was developed that minimizes the need for such prediction algorithms.

(4) Keeping computation and control **lags small enough** so that the positioning system had sufficient time to position the touchboard was a fundamental systems engineering task required careful accounting of each time lag in the system. Continual refinement of the timing budget allowed early identification of problems. Computational problems could be treated by using dedicated board level processors for the control algorithms, by microcoding key computations, and by using interrupt-driven synchronized event processing.

(5) Providing redundant **safety** systems to protect the operator during development and use was considered to employ software to ensure the positioning system is commanded to stop before the tracked hand moves into the motion space, an independent light curtain electronic system that directly shuts down the system upon any intrusion into the motion

space, and mechanical guards around the working mechanisms to ensure than intruding elements were deflected rather than caught or pinched.

The identified technical risks made the Phase II implementation a major systems engineering challenge. Along with the direct risk of meeting the technical objectives was the associated risk of keeping the project on schedule and within budget as the various challenges were faced. The results are presented in the following two chapters.

3. System Concept

A traditional flight simulator is built using a replica of the cockpit of the aircraft being simulated. Building a replica cockpit is expensive, as a different replica cockpit is needed for every type of aircraft to be simulated, and it is difficult to keep up with changes made to the real aircraft. Conceptually, it would be better to have a virtual cockpit in which the elements of the cockpit are determined entirely by software. Then the expense of constructing physical replicas could be saved; one simulator could be used for many different types of aircraft, and after the simulators are in service the simulators could be quickly updated to reflect modifications in the real aircraft.

For a virtual cockpit, the appearance of a cockpit can be represented by computer generated imagery on a head-mounted display (HMD) worn by the user. The fidelity of this approach is limited by the resolution of the HMD and by the realism of the computer generated imagery for the display. HMD technology and image generator technology are such that the best currently available technology is probably barely acceptable for the application, and even then at relatively high cost. However, current trends toward lower cost and improved performance should close the performance gap considerably within a few years' time.

In addition to a visual simulation, a virtual cockpit also needs a simulation of the force and tactile sensations of touching the controls. The controls include the primary controls and the instrument panel controls. The primary controls are the joystick and rudder pedals or their equivalents for steering the aircraft. The instrument panel controls include switches, knobs, push buttons, and keypads. Replica controls could be provided to be used with the simulated imagery, but doing so would not meet the objective of having a simulator that is reconfigurable in software.

For the prototype virtual cockpit discussed here, replicas were used for the primary controls, but a software reconfigurable approach was adopted for the instrument panel controls. Because the simulator user is wearing a head-mounted display, and because the user touches only one instrument control at a time, it suffices to present to the user only the single control being touched. This is accomplished by using a collection of about a dozen different types of physical replicas of controls, and putting the correct type into the correct place to be touched whenever the user actuates a control.

To select the correct type of control and put it into place, the user's hand and fingers must be tracked and the positions extrapolated forward to determine which control will be grasped. A robotic mechanism then quickly puts the correct type of control into place in time to be actuated. A user may believe that different toggle switches are being flipped at different places on the instrument panel, but in fact the same toggle switch is being touched in all the different positions. A mechanism must be provided to put the switch in the correct "up" or "down" position while the switch is being moved to a new position. Similarly, rotary controls must be brought into correspondence with the way each control appears in the user's HMD imagery.

For the concept to be practical, the few replica controls must be moved rapidly to stay ahead of the user's hand motions. The requirements were quantified by analyzing cockpit videotapes taken in flight and also videotapes taken in a lab setup. In the lab, a number of non-pilot subjects were videotaped as they actuated switches and knobs in a prescribed sequence. Timing requirements were determined by stepping through the videotapes frame-by-frame and recording the times required to reach the controls. The derived requirements were that the controls must be repositioned with an acceleration of up to four g's and a speed of about three meters per second. Maximum acceleration and deceleration are required when closely-spaced controls are actuated in sequence.

3.1 System Configuration

The system is designed with three major subsystems, one each for robotics, tracking, and visual simulation [Fig. 3.1-1]. Each subsystem is controlled by its own computer, with communications links transferring data among the three control computers.

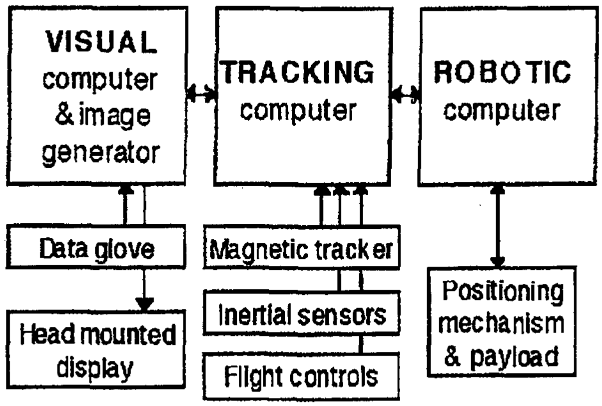


Figure 3.1-1 Three major subsystems.

The tracking subsystem is built around a personal computer running the QNX real time operating system [Figure 3.1-2]. The tracking computer interfaces with the hardware that measures the position and orientation of the user's head and right hand and runs software that filters and extrapolates the tracking data. It determines which switch the user is about to actuate and sends commands to the robotics subsystem to move the selected switch into place. It keeps track of the orientations to which the knobs and toggle switches are moved. It also interfaces to the user's flight control joystick and throttle and computes the position of the simulated aircraft. The tracking computer sends the positions and orientations of the head, hand, and switches to the visual simulation subsystem, which in turn generates imagery for viewing in the user's HMD.

The robotics subsystem includes a VME-rack with a control processor and interfaces, servo power supplies and amplifiers, and power distribution circuitry. The VME-based control processor receives high level commands from the tracking computer over a 38.4 Kb serial interface. The commands from the tracking computer instruct the robotics subsystem to move each of the servo-driven positioning mechanisms to prescribed locations or orientations. The robotics control processor carries out the commands by generating control voltages for each of the servo-motor amplifiers. The motors are equipped with digital shaft encoders and each motor channel is run closed-loop with an update rate of approximately 100 Hz. Each channel is tuned for the inertia and spring constants associated with the channels' hardware.

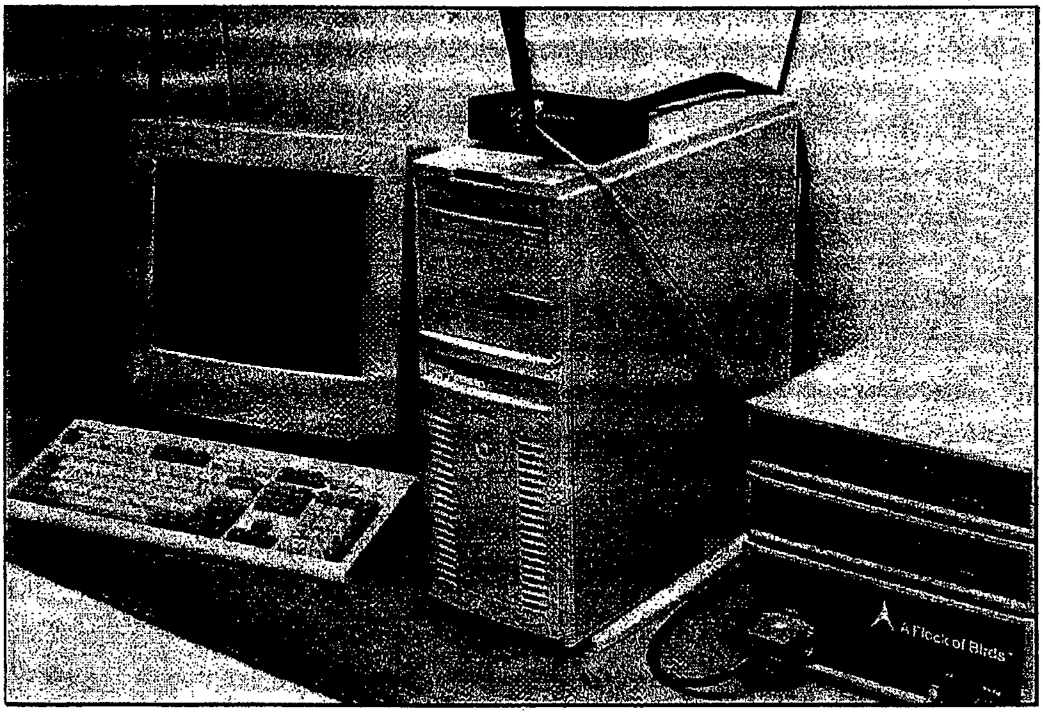


Figure 3.1-2 TOPIT tracking computer, magnetic tracker electronics (right) and HMD electronics (on top of computer).

The visual subsystem is built around a Silicon Graphics Onyx computer having a RealityEngine2 image generator. The visual computer receives data from the tracking subsystem over a dedicated Ethernet link having less than one millisecond latency. The visual computer has a database of polygons modeling the cockpit interior, the user's hand, and terrain outside the simulated aircraft. It assembles the scene from the polygon models, putting each model in its correct relative position. A dataglove worn by the user provides the positions of the fingers directly to the visual computer.

3.2 Tracker

Magnetic trackers are commonly used in virtual reality systems. They use compact, lightweight sensors, are unencumbering, measure all three position coordinates and all three orientation angles, and are economical. The limitations of magnetic sensors are that metallic objects distort the tracker fields thereby producing static errors, they are susceptible to interference from electrical noise sources, and there tends to be lags in the measurements. The lags come from filtering the noise inherent in the measurements. In many applications, none of the limitations prove severe. For the virtual cockpit, however, the tracking could not lag significantly and must work in the presence of the metal and motors of the robotic positioning device.

One alternative to magnetic tracking was mechanical tracking. A mechanical tracker uses stiff rods connected by joints having encoders. Mechanical trackers are low cost, extremely accurate, immune to noise, and have no appreciable lag. Unfortunately, mechanical trackers are encumbering since they require a mechanical linkage to the users head or hand. They are best used when the space of possible motion is small, and might be acceptable for head tracking a seated user. For hand tracking in a virtual cockpit, the encumbrance would not be acceptable in the long run. Nonetheless, mechanical tracking could be a backup method, at least for lab evaluation of the virtual cockpit.

There were a number of optical tracking systems available. These systems use a variety of principles for tracking. Some use high resolution cameras tracking reflective markers. Others use sensors that detect a scanning infrared laser. Optical tracking systems are typically so accurate that the orientation of a surface can be computed by tracking three points on the surface. Optical tracking would be a good choice for a virtual cockpit, but the cost of commercially available systems ruled it out for the prototype.

The alternatives were to work with the limitations of magnetic trackers or to attempt development of a low-cost optically-based tracking system. We opted to work with the magnetic tracker. To minimize magnetic field distortions, the robotic mechanism would have to be made from non-magnetic material. Aluminum was tested and found to be nearly as bad as carbon steel in inducing tracker distortions; it apparently induced distortions in the electric field component of the tracker transmission. The best metal was non-magnetic stainless steel (series 300), so that was preferred for construction. Wood or plastic might have been used, but the structure could not be made acceptably stiff.

As it turned out, the distortions due to the metal structure were up to about 4 cm of error, which could be reduced substantially by calibration and look-up tables. The goal was to provide overall tracking accuracy of about 5 mm, which seems achievable.

To treat the problem of tracker lag, an inertial sensor package was added to the magnetic hand tracker. The package initially consisted of three miniature accelerometers and three angular rate sensors. This inertial package was larger than desired, about three inches square and an inch think; however, it could be mounted on the forearm rather than on the hand itself.

The alignment of the axis of each sensor was required to be orthogonal in order for the software to receive correct information. This was not attainable with the aforementioned setup, so two replacement sensors were purchased – a triaxial rate gyro and a triaxial accelerometer. This new inertial package was slightly more compact and could be fitted on the user's wrist.

Combined with inertial data, the magnetic tracker data could be smoothed with only about a fifth of its typical lag, roughly 30 milliseconds rather than 150 milliseconds. Also, the accurate velocity and acceleration measurements enabled better extrapolation of the hand position. Extrapolation is required to compensate for delays of 30 to 60 milliseconds in the image generator, and to extrapolate the hand position to determine which switch is selected.

The magnetic and inertial tracking data are combined in software using Kalman filtering, a technique often applied in multi-sensor navigation systems. The computational requirements of the filter are just within the capabilities of a 200 MHz personal computer, although they could be reduced with more optimization.

3.3 Robotic Mechanism

The starting point for selecting a robotic mechanism was to consider off-the-shelf devices such as industrial robots. The robot must position a payload having an assortment of controls together with the motors necessary to reposition the rotary controls and toggle switches. An initial estimate was that the payload would weigh about five kilograms, although the ultimate design totaled about eight kilograms -- a consequence of the stainless steel construction.

Industrial robots were available which meet the requirements, but they are large, high powered, and expensive. Industrial robots are designed to have a long reach into a large workspace, and consequently are built with heavy links which in turn must be driven by powerful drive mechanisms. Cockpit instrument panels are wide and fairly high, but the panel surface does not encompass much depth. A custom robotic device was designed to take advantage of the restricted workspace. It cost less and is safer than an industrial robot.

The large reach of the industrial robot would have posed a safety problem. Potentially, the robot could move respectable masses at high speeds into the space of the user. Since it would not be acceptable to operate only with software limits the robot would have to be physically modified to make it impossible to travel into the user's space. The customization required would further added to the cost of the device.

Finally, industrial robots are not typically made of non-magnetic stainless steel. Making a new device permitted constraining the design to be compatible with magnetic tracking. In a new design, the electric motors could be positioned as far away from the trackers as possible.

The manipulator design recalls some of the design features of an old-fashioned pen plotter [Figures 3.3-1 and 3.3-2]. The horizontal and vertical axes are driven by Kevlar^{TM} cables. Using cables for both drive mechanisms avoids making the outer axis motor bear the burden of having to move the inner axis motors. Both major axis drive motors are affixed to the frame, one on either side, near the ground, and back from the trackers. A relatively small motor, which moves the payload in and out, is carried with the payload. The electronics cabinet, which houses the servo electronics and system power control and safety circuitry, can be seen to the left of the user [Figure 3.3-2].

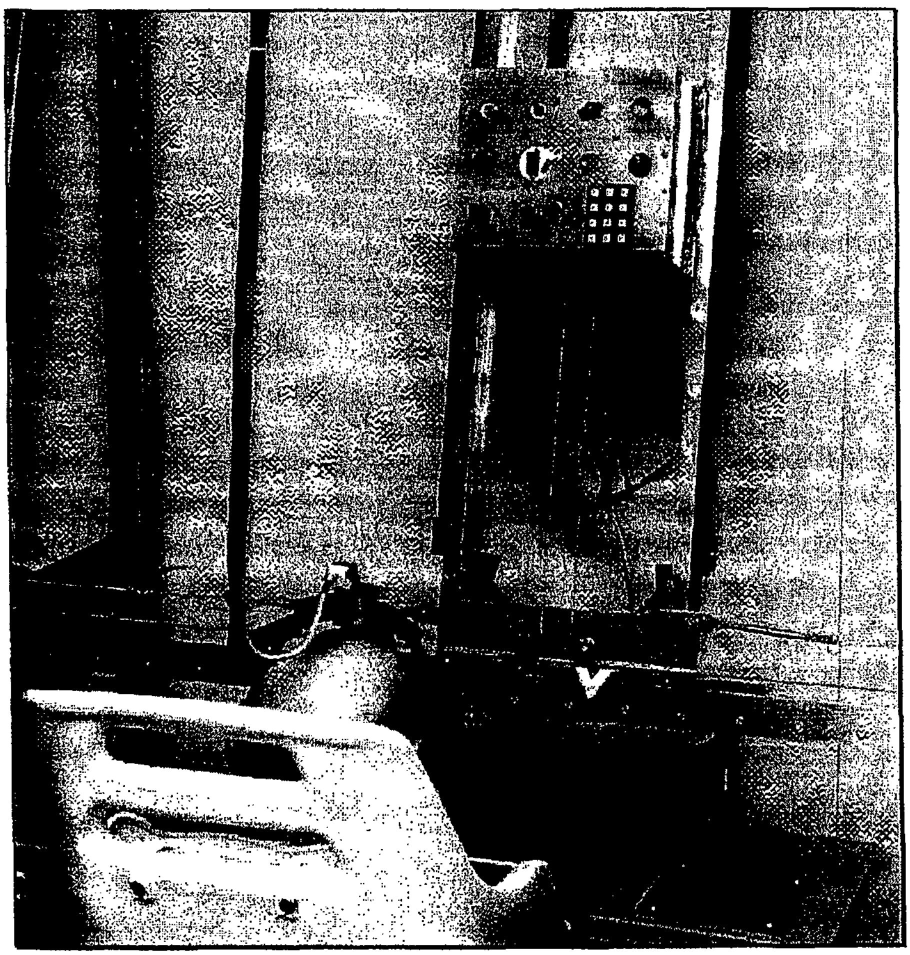


Figure 3.3-1 TOPIT system showing operator's station, X-Y manipulator, and touchboard.

Figure 3.3-2 The final design uses cables to position the switch payload in x and y axes.

The main design feature for safety is constraining the user and the robotic mechanisms to their own workspaces. The user must cross into the robot's workspace to touch the payload controls, but the hand is tracked and the software is designed to bring the mechanism to a halt before the hand crosses into the mechanism area. Still, one must account for possible software failures, for untracked parts of the user's body, and for bystanders. These additional safety provisions are discussed in section 4.5.

In the current design only the head and the right hand are tracked. Tracking the left hand is mainly a cost issue, and doing so would allow controls to be actuated with either hand as well as enhancing safety. The untracked left hand is required to be kept on the throttle. A switch on the throttle must be continually depressed; if it is released the mechanism halts. The throttle switch tends to keep the user properly seated away from the mechanism. A second switch could be added to the seat back to further ensure the head is kept back from the mechanism; leaning forward would release the seat switch and stop the mechanism.

The payload [Fig. 3.3-3] moves with maximum speed about equal to a hand moved laterally to activate a switch. This is not fast enough to cause a serious injury if, due to a system failure, it were to hit the user's hand in motion. A potential danger lies in pinch points, where the users hand might be caught in a closing space between the frame and the payload or traveler. Pinch points are prevented by making the frame oversized and mounting rubber blocks to stop mechanical travel short of the frame.

An emergency stop circuit is included in the design. This circuit is hardwired to a single relay that disconnects and then short-circuits the drive motors, quickly bringing the mechanism to a halt. When the virtual cockpit is in operation, an observer can actuate one of two emergency stop switches if the user or a spectator gets too close to the mechanism.

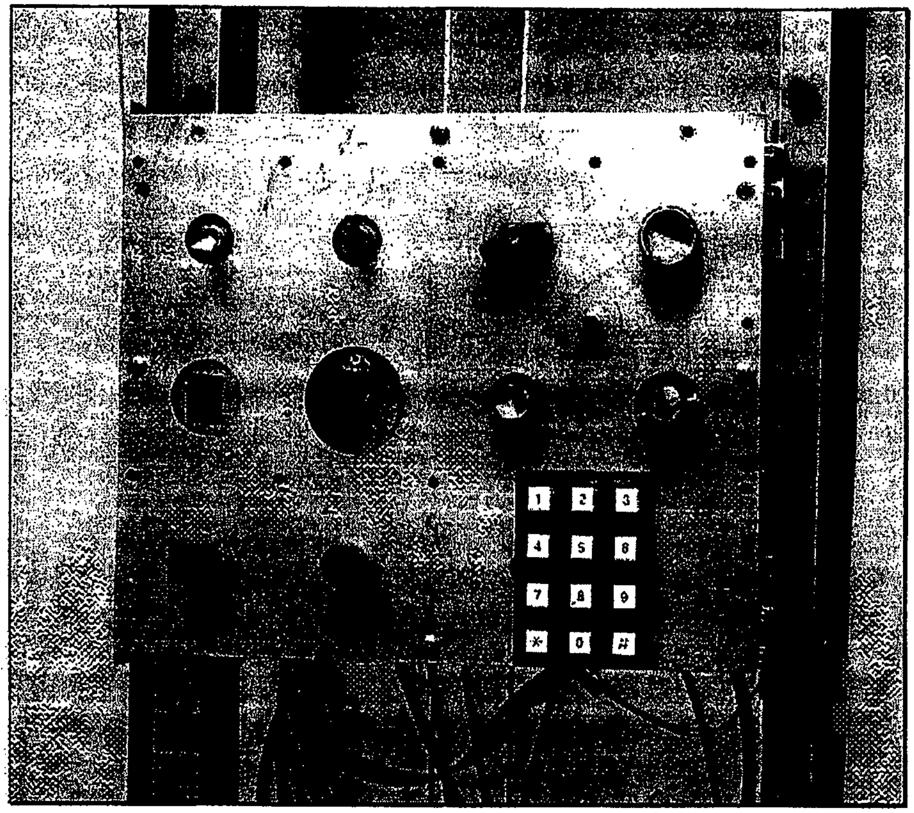


Figure 3.3-3 The payload includes switches and knobs which are rotated to the needed orientation.

Covers would be added to any production device to prevent a bystander from reaching any of the drive mechanisms from the sides or rear of the device.

3.4 Visual Simulation

The Onyx RealityEngine2 computer uses position data from the tracker to prepare the visual scene from pre-stored polygon models of the cockpit, the user's hand, and the out-the-window terrain [Fig. 3.4-1]. The Onyx computer runs a real time version of UNIX in two processors, and we wrote the visual simulation using Silicon Graphic's Performer application package.

There is a delay of one to two video frames in generating the image, marked from the time position data arrives in the tracking computer until the generated image is displayed to the user. The image is generated to correspond to where each moving element of the scene is expected to be at the time when the image appears. Consequently, the position and orientation of every moving element in the scene must be extrapolated forward from the time at which the position and orientation of the element were measured to the time at which the image appears. Simple extrapolation using velocities and accelerations works adequately for times up to about 100 milliseconds.

The imagery is presented to the user on a head-mounted display. Separate images are computed for each eye to provide true stereo. The user's judgment of his hand position relative to the instrument panel is helped significantly by having stereo imagery.

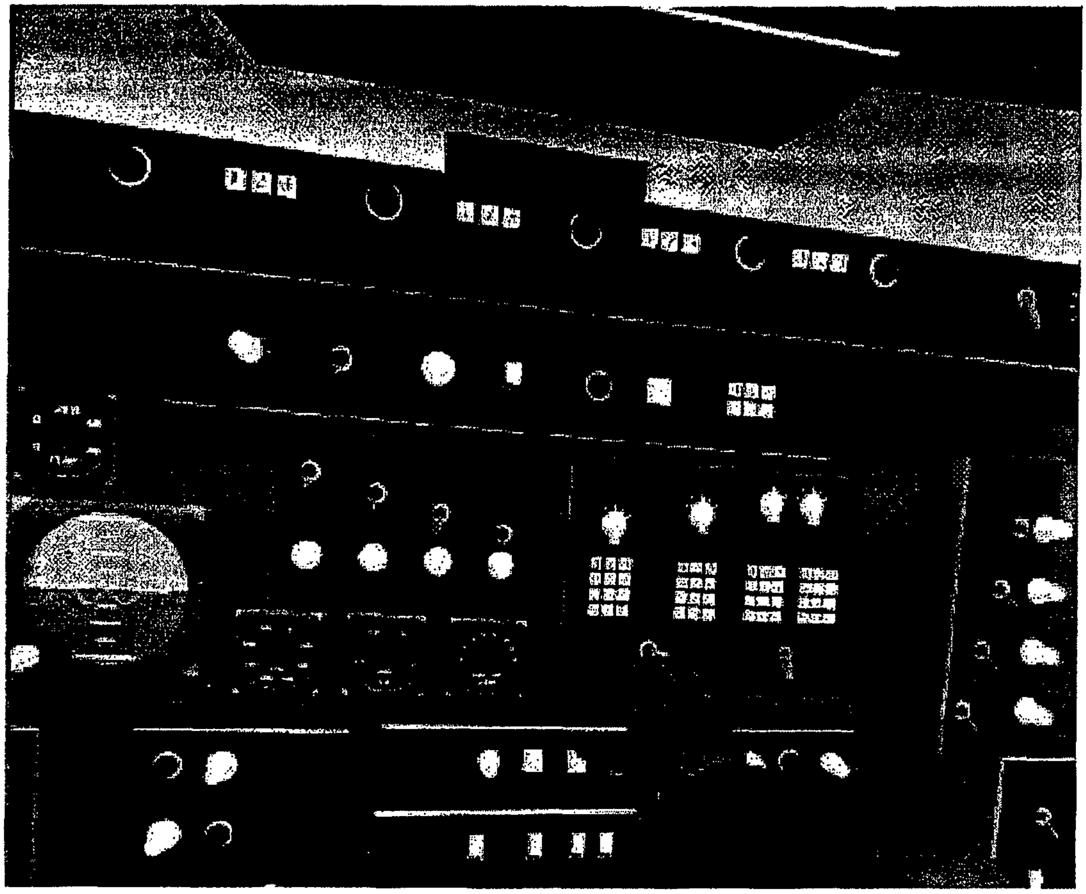


Figure 3.4-1 Generated imagery includes the user's hand.

A moderately priced liquid-crystal based head-mounted display is used, the Virtual Research VR-4. This HMD provides a resolution of about 320 x 480 pixels. This resolution is adequate to see and grasp the controls, but it is not sufficient to read either normal-sized control labels or many instrument panel displays. The compromise in resolution was forced by the economics of the prototype. High resolution head-mounted displays are expensive, and the increased resolution requires more expensive image generation capability. In the development system, the emphasis was on demonstrating the feasibility of the force and tactile feedback mechanism rather than the display.

The polygon processing capacity of the image generator (about 220K polygons per second) limits the scene complexity. The use of stereo imagery cuts the scene complexity in half relative to what it would be otherwise. The cockpit interior is inherently complex, with the knobs and switches modeled in three dimensions. The desired frame rate is at least 30 frames per second, and the allowable polygon complexity per frame is lowered in proportion to the frame rate.

The technology of HMD's and image generators is advancing rapidly, so that these system elements are expected to be less of a limiting factor in the future. The possibilities of advancing technology was part of the reason for partitioning the tracking and visual simulation subsystems around separate computers. The partitioning was designed to simplify substitution of lower-cost image generation technology without having to recode the PC-based tracking computer software. The partitioning could also help in making the hybrid magnetic and inertial tracker with its Kalman filter into a separate PC-based subsystem for other applications.

3.5 System Integration

The system as described is currently working well enough to demonstrate the feasibility of the concept, although there are improvements to be made. A few of the lessons learned in system integration can be cited.

Initially, the software was designed so that the payload would be moved to mirror the position the hand until the hand was within a few inches of the surface of the virtual instrument panel. When the hand reached the pre-determined close distance, the software would pick the switch to be grasped, put the switch into place, and freeze the payload position until the switch was actuated and the hand withdrawn.

The error in this design soon became apparent. The robotic system is designed to accelerate the payload at up to 4 Gs. When the hand was still distant from the panel, the attempt to mirror the position of the hand with the payload produced a great deal of pointless violent motion of the payload. The cure was to put an extra filter on the position data given to the robotic subsystem. The extra filter has a time constant which is adjusted to provide smoother position following when the hand is further from the virtual instrument panel.

Another unanticipated problem was resonance in the mechanical frame of the positioning mechanism. The total weight of the traveler mechanism and the payload is approximately 44 pounds. When accelerated at 4 Gs, the resulting reaction force is therefore about 176 pounds. The frame is welded from 2 inch square stainless steel tubing and is very stiff.

However, tracking movements of the hand produces frequencies which excite resonances in the structure. When resonating, the deflections at the corner of the structure may be as much as a centimeter. It is not clear if the deflections actually degrade system performance, but the fear is that they affect the servo control loops. The shaking is also disturbing to bystanders. Custom pampers were added across the diagonals of the structure to dissipate the resonant energy.

A shell, floor, and an integrated pilot's seat were added to the frame for aesthetic reasons. A compartment underneath the pilot's seat was also built to house the sensor electronics.

3.6 Future VR Systems/Phase III Applications

Forethought in system partitioning and timing analysis, as well as making tradeoffs among the limitations of subsystems are central to good virtual reality systems design. Systems with human/robotic interaction inevitably pose serious safety considerations. With present technology and experience, it is feasible to build a limited class of virtual reality systems, such as the virtual cockpit, to provide force and tactile feedback.

Robotic positioning systems

The concept of robotic positioning of touched objects is not a universal solution to the problem of providing force and tactile feedback. It applies when there is a limited class of objects to be touched, when fidelity is important, when the simulation of external forces is important, and when safety constraints can be met. Alternative methods include special gloves having air bladders or other touch stimulating transducers, exoskeleton devices attached to the hand or body, and robotic devices continuously attached to specialized tools for simulating the forces encountered in the tool use.

Within its realm, the method of positioned objects does offer interesting possibilities. Note that in the virtual cockpit, the system could easily provide the capability for touching the window glass or the flat surfaces of the cockpit surfaces. In an architectural walk-through or entertainment system, the user might touch various surfaces of the environment, presented by a robot having a selection of surfaces. The performance requirements for the robotic mechanism in a walk-through environment are different from the virtual cockpit. The larger workspace might dictate something closer to an industrial robot design, but larger surfaces and slower user motion could relax the speed requirements and make the system safe.

A robotic system could also initiate contact. Consider a virtual reality entertainment system with a "dungeons and dragons" scenario. A user wearing a head-mounted display explores dark passageways. At a critical juncture, the user hears a sound from behind and at the same time a robotic device reaches out with a rubber finger to deliver a poke in the ribs. There is potential for such systems.

Hybrid tracker

The hybrid filter could be commercialized by itself. It would need to be smaller and lighter weight. The hybrid sensor module uses two types of sensors; accelerometers and angular rate sensors. The accelerometers available today seem suitable but the angular rate sensors are bulky and relatively heavy. For the hybrid tracker to be smaller and lighter alternative angular rate sensors are needed. There are several sensor companies already working on better angular rate sensors but suitable products are not currently available - perhaps in a year or so.

A commercialized hybrid tracker, along with the Kalman filter software we developed, would make a great add-on to commercially available magnetic trackers produced by Ascension and Polhemus by providing an accurate low lag tracking system. Many aerospace and commercial applications would benefit from such enhanced performance.

4. Phase II Results

W have accomplished all of our objectives. A picture of the final system and snapshots of subsystems and components is presented in section 1. Detailed evolution results of each subsystem, compared against the objectives, are described in the following.

4.1 Positioning System

Project Objective #1: ***Build a positioning system that moved fast enough but without excessive size, power, or cost.***

The positioning system includes the X, Y, and Z axes of motion. To move fast enough to keep up with anticipated user motions, the TOPIT™ had to accelerate at 4 Gs and move at 100 inches per second in both the X and Y directions. The rates were achieved in both axes even though the weight of the payload exceeded expectations. Two oversights, a miscalculation of the drive drum diameter and the effects due to gravity, previously prevented the Y axis from attaining this specification. The incorrectly sized drive drum was corrected by designing, fabricating, and installing a larger diameter drive drum. The effects due to gravity were compensated by using a coil spring attached by one end to the frame and the other end around the new drive drum.

A significant technical achievement in the X, Y motion control was the successful implementation of motion control software which not only controls the servos in such a way that the payload is transported to its commanded location in minimum time without overshoot but also coordinates X and Y motions such that straight-line X-Y motions are achieved. Straight-line motions minimize the demand on the servo power supply by reducing the commanded acceleration and speed on the axis with the shortest distance to travel such that the X and Y motions take the same amount of time to complete.

We incorporated a variable gain filter on X/Y motions so that the servo system responds to hand motions more slowly when the user's hand is more than a few inches from the payload. The filter is progressive; the further away the user's hand, the slower the response. Highest performance moves (high speed and high acceleration) are only needed for final payload positioning when the user's hand is approaching the payload. General tracking is all that is required otherwise.

To achieve X and Y drive performance and minimize the required power and cost it was necessary to minimize the weight which had to be moved. The Z drive motion is attached to and moves with the payload by the X and Y drive motors. In an effort to save weight and achieve desired performance in the X and Y directions a motor was chosen for Z drive. Initially, we had some mechanical difficulty in setting up the Z drive correctly. After a redesign of the linear bearing and ball screw assembly, the Z drive now works well; although, the maximum travel is limited to four inches rather than the specified six inches.

Since we planned to build a virtual aircraft cockpit the overall size of the TOPIT manipulator frame was dictated largely by the application. One design goal of the project was to have the ability to simulate an aircraft cockpit dashboard area 42 inches wide by 30 inches high. We achieved horizontal (X direction) excursions of 42 inches but safety concerns and the need to allow emergency stopping areas above and below the payload limited vertical excursions to about 22 inches - still sufficient for proof-of-concept testing. We believe the overall size of the device could be made somewhat smaller by repositioning turnbuckles used to tension the X and Y drive cords and by repositioning several of the pulleys used by those cords. A somewhat different payload and touchboard design where the touchboard (the switch panel on which the user controls are located) is partially cantilevered would increase the vertical excursion without compromising safety.

The following subsections describe the evolution of the system. While the various efforts are discussed sequentially, the reader should note there was considerable overlap in these efforts.

4.1.1 Desktop prototype hardware development and testing

To confirm our motor calculations, verify our bearing choice, and to provide a means of testing the magnetic tracker compatibility we decided to build a single axis desktop prototype on which performance with loads of up to 25 pounds were tested. The prototype used a custom workbench-like structure that was constructed of wood so that it would not produce magnetic interference. Initially the tracks for the wheels were made of steel. We also tried aluminum and stainless steel tracks later. Testing suggested we needed to use non-magnetic (series 300) stainless steel for the proof-of-concept system. The load was moved using low-stretch Kevlar cords and a series of pulleys connected to a drum which in turn was direct-coupled to a two-horsepower servo motor.

We originally planned to use v-groove wheels and track. The quality of the wheel bearings and the wheels themselves turned out not to be satisfactory so we used cam followers, which have heavy duty roller bearings, for the wheels and aluminum channel for the track [Figure 4.1-1]. Note the turnbuckles which are used to tension the Kevlar cords.

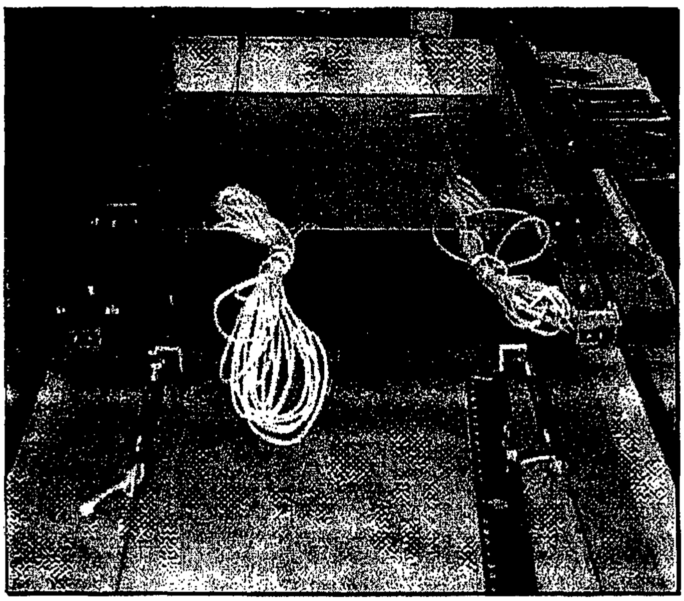


Figure 4.1-1 Desktop prototype with cam followers used for wheels and aluminum channel used for the track.

The drive mechanism [Figure 4.1-2] was located underneath the desktop. Note the hand crank, which was used for initial testing, the drive drum, and the Kevlar cord wrapped around the drum.

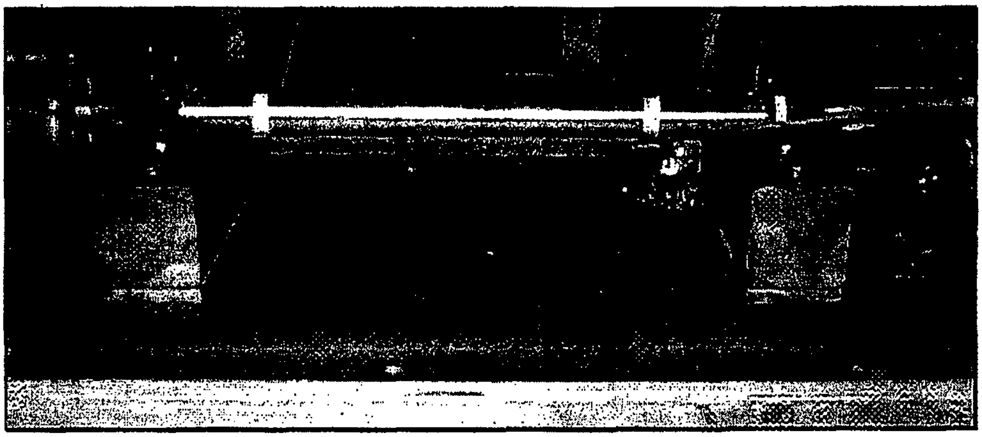


Figure 4.1-2 The desktop prototype drive mechanism.

After receiving all the necessary components and completing construction of the desktop prototype it was wired to the servo electronics and testing began. The safety system was verified to be working correctly. The homing sequence was programmed and verified; the homing sequence is the process by which positioning is automatically calibrated by moving the payload so as to actuate switches at each end of the device. We also determined the sliding carriage could be positioned with an accuracy of 1 mm or better over its travel - more accurate than needed.

The ultimate requirement for the system is to provide 4 Gs of acceleration and 100 inches per second velocity with a payload of 25 pounds. A piece of scrap iron was used as the load on the desktop prototype [Figure 4.1-3].

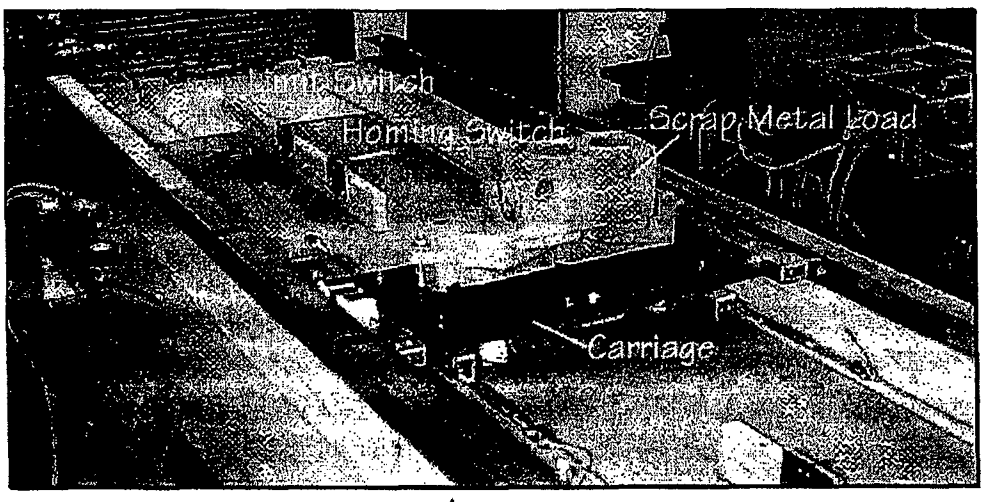


Figure 4.1-3 Desktop prototype with scrap metal load.

The test envelope was first expanded to 1 G of acceleration and 100 inches per second velocity without additional payload weight. As testing progressed we identified and cured the various problems that surfaced. Many of these problems would have occurred later on the proof-of-concept system had we not first discovered them on the desktop prototype. We therefore believe the desktop prototype effort to be worthwhile since it uncovered problem early in the program allowing more time to correct them before construction of the proof-of-concept device. Several examples of problems encountered follow.

First, there was more friction in the unloaded transport mechanism than we expected. We changed the cable pulleys to a larger type with better bearings and that helped, but did not reduce the friction to our expectations. The motor is powerful enough to easily overcome the transport friction, but we still wanted to reduce it further and experimented accordingly.

Second, servo power amplifier shut itself down short of providing all the power we needed to achieve 4 Gs of acceleration. We decided the protection limits were set too low, so we expanded them without risking the amplifier. We later upgraded the amplifier to allow it to handle greater loads.

Third, there was some concern with the motor starting to heat up during continuous operation. This turned out not to be a problem.

We were concerned that Kevlar cable used in the drive transport might creep or be too stiff but later concluded the Kevlar cable worked well.

We observed motion oscillation before settling at high loads. This was cured by correctly tuning the motion control software to account for the "as built" mechanical stiffness of each axis of motion. Such tuning is typical of a servo systems and Delta Tau (manufacturer of the PMAC motion controller card) provides software specifically designed to permit such tuning.

Test objectives for the desktop prototype were established and software written to support the testing needed to accomplish the objectives. Desktop prototype test objectives and test results are discussed below:

1) Verifying operation of the limit-switch safety system

The limit switch system worked reliably and correctly.

2) Verifying the homing and position calibration sequence

The homing and position calibration software worked reliably and correctly.

3) Identifying sources of friction and loading in the transport mechanism

Excess friction was traced to the bearings in the cable pulleys. Higher performance pulleys were installed, and this significantly reduced the friction.

4) Establishing ways of checking and maintaining component alignment

A four-turnbuckle system was installed and has proved suitable for alignment. However, the approximately six inch length of each turnbuckle reduced the usable active area of the desktop prototype by almost a foot. A more compact turnbuckle mechanism, where the mechanism overlaps with the width of the payload carrier, was designed and installed. It worked fine and provided about twelve additional inches of payload travel.

5) Checking the positioning accuracy and repeatability

The positioning accuracy is better than a millimeter, and we had no problems with the cable stretching or slipping on the drum. We need only about 3 - 5 millimeters accuracy.

6) Establishing the acceleration and velocity limits of 4 Gs and 100 inches/second

After a few problems with amplifier shut down and amplifier failures we achieved the design goal of 100 inches per second for large excursions of the manipulator. We also set a second design goal of 4 Gs for small excursions. Both of these goals were met with a 22 pound load - nominally the load to be carried by the "X" axis (horizontal).

7) Checking for design limits, such as potential motor heating, in continuous operation

The payload was cycled for several minutes at 4 Gs acceleration - a time duration far in excess of normal operation. Motor heating was not a problem but the extended duration of maximum acceleration testing caused one of the amplifier IGBTs to fail. The amplifier has subsequently been upgraded and we have not had any other failures.

8) Rechecking all of the design parameters with half and full payload weights

Rechecking the design parameters with half and full loads was completed.

In the process of exercising the desktop prototype several other concerns arose; selection of a slider material suitable for the X and/or Y axes, audible system noise, and Kevlar drive cord stretch. Our efforts in these areas are discussed below.

We tested several types of slider material. We were looking for a durable low friction material. We found that Teflon-loaded Delrin sliding against a stainless steel track worked reasonably well. The concern was that the plastic would gall, which is what happened when we tried nylon against aluminum. However, the Delrin did not gall nor show signs of wear in our tests, even though the stainless steel track material used for the tests was not finished as smoothly as one would like. The final product would have a smooth finished track. Based on the success of these tests, we used the Delrin/stainless steel combination for the y-axis slider in the final design for the proof-of-concept demonstrator.

We noticed the system was noisier than we would prefer when the payload moved. We thought this noise might be distracting to a user even if the user was wearing a headset. We determined the noise was due to the steel payload wheels riding on the aluminum channel we were using as a guide.

The moving mechanism was modified to reduce the noise. The steel wheels running in the channels were replaced with simple flat plastic pads (made from high density polyethylene). It was much quieter in operation and potentially lighter weight. We were concerned about potential wear and monitored it as testing continued - the plastic surface showed some galling after operation. Pads were required facing each of the three sides of the channel to keep the slider from twisting under high accelerations. There was more friction with the plastic than with the wheels. Spraying the inside of the aluminum channels with silicone lubricant reduced the friction considerably, but it had to be sprayed fairly often - something like once a day.

Tuning of the servo loop control software was best done with a payload which moved smoothly with uniform resistance to motion. Since the temporary slide mechanism did not provide smooth motion, the payload was outfitted with a wheeled carriage mechanism which was constructed using nylon ball bearing wheels to which rubber O-ring tires were attached [Figure 4.1-4]. While it provided the desired smooth (and quiet) operation of the desktop prototype, it was too bulky for use in the proof-of-concept demonstrator.

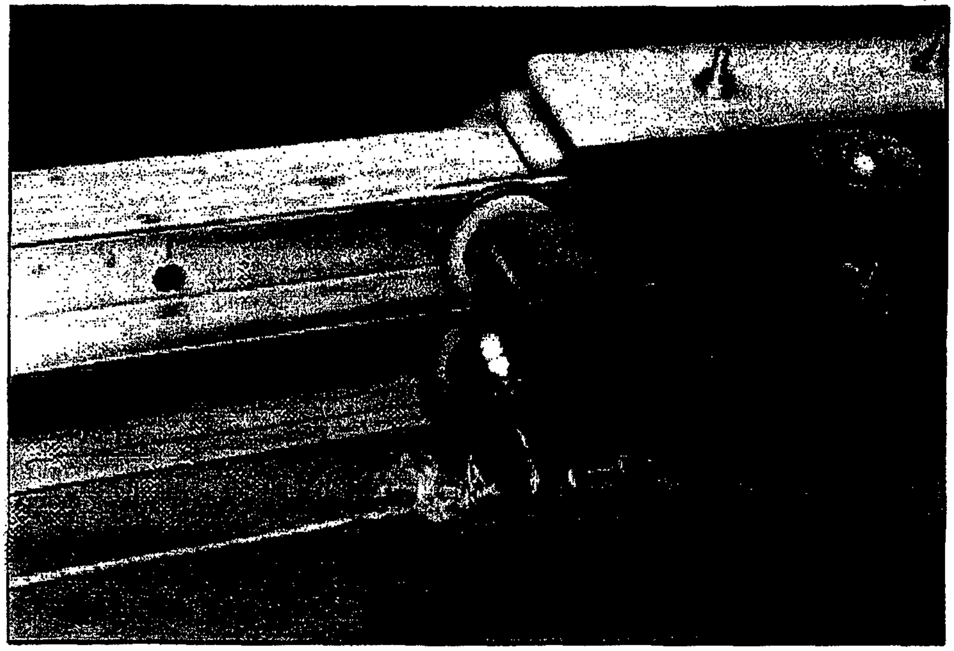


Figure 4.1-4 Portion of the payload and wheeled slider assembly - rubber "tires" on the wheels grip a flange to provide quiet operation.

The desktop prototype was later retrofitted with PTFE slider material (a high-performance plastic material). It was be used in conjunction with stainless steel rails which provided smoother motion than the aluminum rails used with the first set of sliders. The galling we witnessed with the first sliders was also eliminated.

Note that performance of the original steel wheeled arrangement was fine except for the noise generated. Noise is mainly a cosmetic issue. Nonetheless, using plastic sliders had the additional benefits of being lighter and simpler than either of the wheeled arrangements, as well as generating less noise.

We noted we had to tighten the Kevlar™ drive cords periodically. While not a major problem, we looked into the cord stretch problem by conducting some tests. We determined that when in constant use, the cords stretched slightly due to heating. The result was lower than desired cord tension. We also determined that as the cords cooled, they returned to proper tension. We do not believe this condition will exist when used in a true simulation environment where the TOPIT is exercised much less frequently than in our testing scenario. We therefore decided not to take any additional action other than to continue to monitor the situation.

4.1.2 Manipulator design

We define the manipulator as including the X, Y, and Z drives and related hardware. We designed the TOPIT manipulator based upon lessons learned with the desktop prototype. We decided to stay with the cable-driven mechanism used for the desktop prototype, rather than switching to the originally-proposed belt-drive arrangement. The cable (or "string") drive worked fine on the desktop prototype and has the advantage of keeping the servo motor further away from the tracker. We also decided to use cable drive for the y-axis, i.e., the vertical axis [Figure 4.1-5]. This saved the x-axis motor from having to move a y-axis motor. The y-axis motor remains stationary. The y-axis motor is coupled to the Y slider through pulleys.

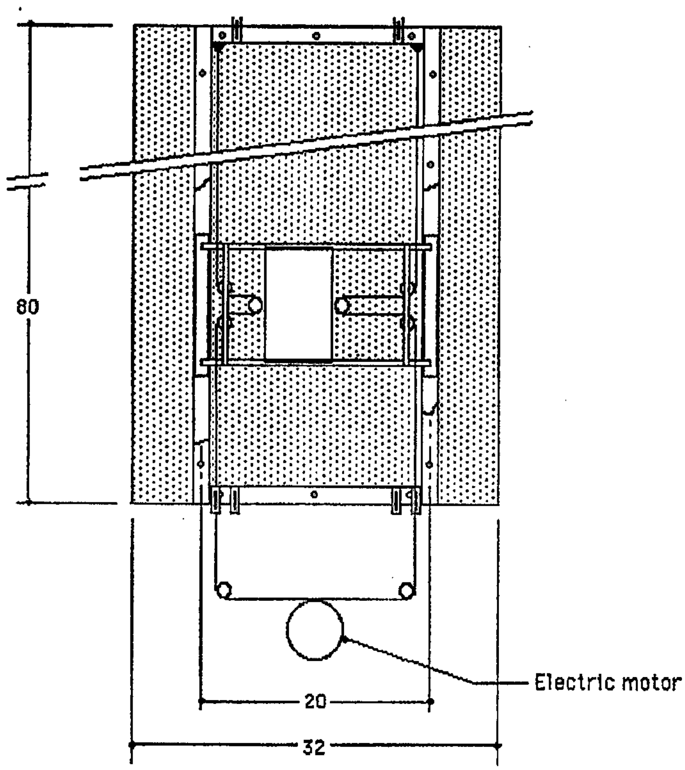


Figure 4.1-5 How a stationary motor drives the y-axis independent of x-axis motion.

We combined the operator station with the manipulator frame design to reduce the overall complexity. By adding brackets to the manipulator frame to hold the operator seat which in turn held the operator hand controls (joystick and throttle) [Figures 4.1-6 and 4.1-7] we eliminated virtually all of the mechanical design effort from the operator station. Since the position of the operator's seat and hand controls are fixed to the manipulator frame, they do not have to be tracked during real time simulation. Note the hybrid tracker attached to the dataglove [Figure 4.1-6].

Figure 4.1-6 Control station with throttle and joystick; HMD rests on seat.

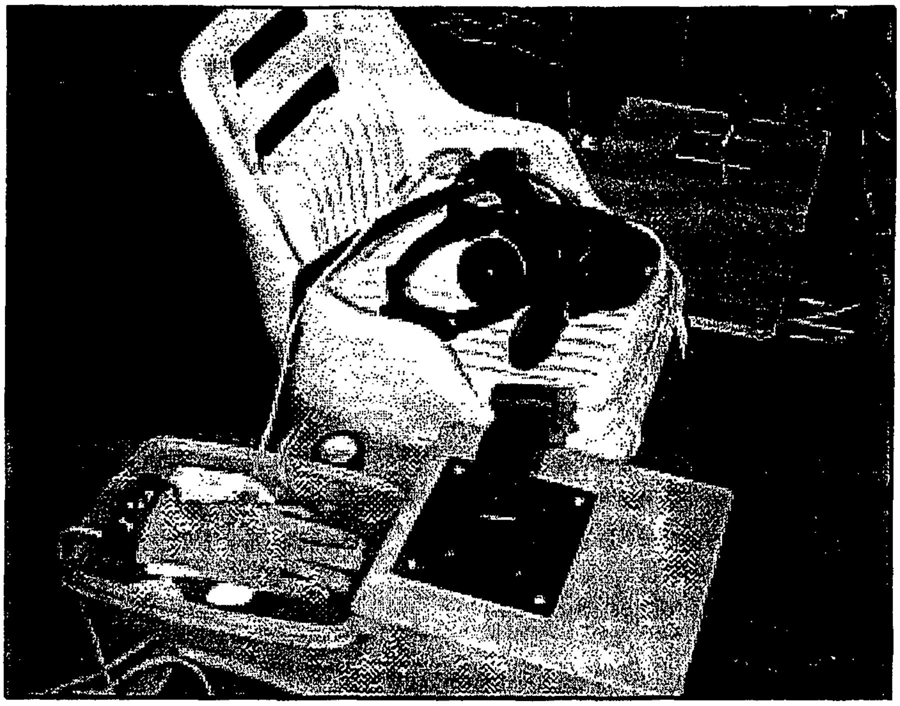


Figure 4.1-7 Control station includes throttle and joystick; HMD rests on seat.

Note the large emergency stop switch to the immediate rear of the user's throttle housing on the left side of the chair [Figure 4.1-7]. The HMD rests on the chair. The dataglove is in a protective box without the hybrid tracker attached.

We decided to make the manipulator with two side A-frames connected by horizontal rails. All manipulator mechanical components were attached to the frame. We used two-inch-square stainless steel tubing for the frame since stainless steel causes much less interference with the magnetic tracker than carbon steel or aluminum. Wood was considered briefly but then discounted since wood is unstable with changes in humidity and it is difficult to produce a sufficiently stiff structure with wood. Stainless steel channels were attached to the frame to support moving parts of the manipulator. Large X and Y-axis drive motors are located at the rear of the manipulator frame to minimize magnetic interference with the user's tracked hand [Figures 4.1-8 and 4.1-9]. Note the drive drums and Kevlar drive cords in both figures.

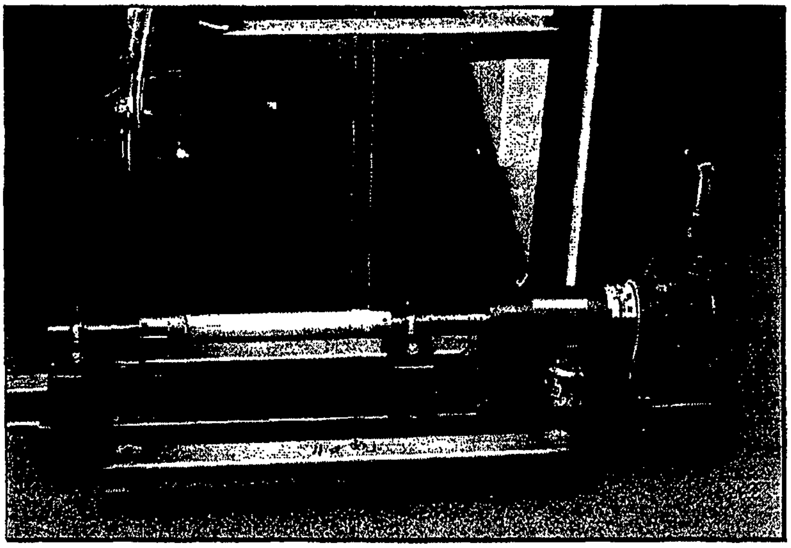


Figure 4.1-8 Lower right side of manipulator frame showing x-axis drive components.

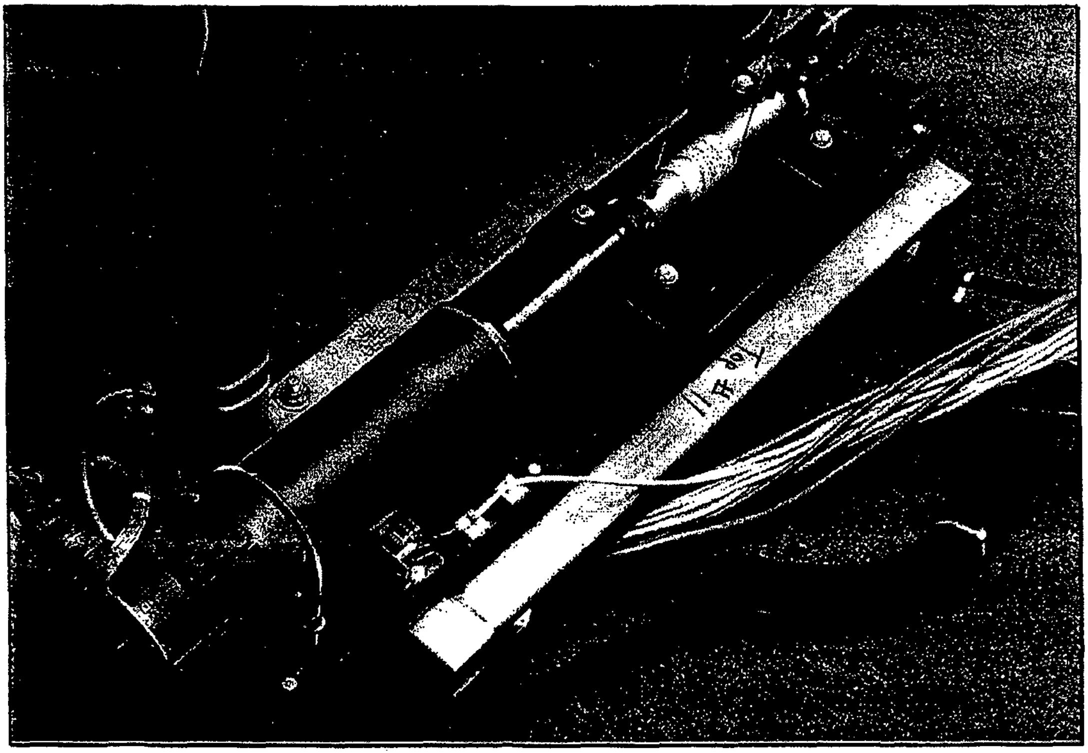


Figure 4.1-9 Lower left side of manipulator frame showing y-axis drive components.

From the drive drums shown above the Kevlar drive cords are routed via pulleys to the payload [Figure 4.1-10]. Cord tension is provided with turnbuckles such as the one shown in the figure.

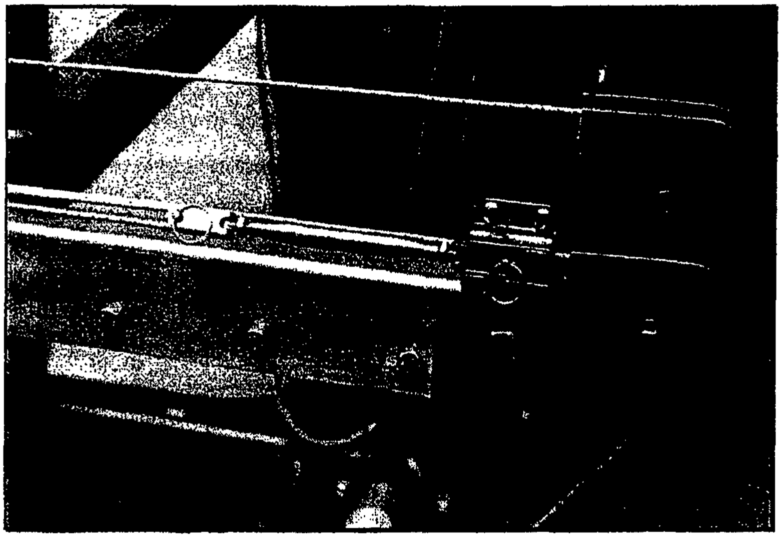


Figure 4.1-10 Manipulator frame showing Kevlar cord, tensioning turnbuckle, and pulleys.

From the pulleys the Kevlar cord is routed to the X traveler (tall vertical frame) and the Y traveler (behind the payload) [Figure 4.1-11]. The X traveler moves horizontally, supported by the manipulator frame at the top and bottom. The Y traveler moves vertically, supported by the vertical side channels of the x-traveler. Graphite composite stiffeners reinforce the x-traveler.

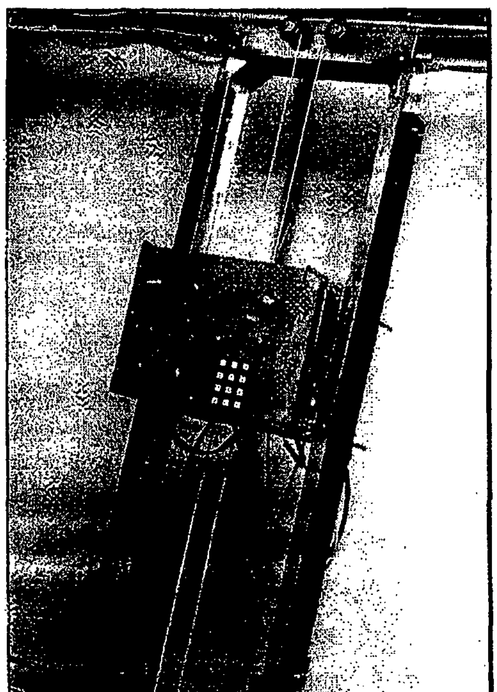


Figure 4.1-11 X-traveler, y-traveler, and payload.

A ball screw drive was adopted for the z-axis. Linear bearings are preferred in applications like this since they provide smooth linear motion but such bearings weigh significantly more and are physically larger than brass bearings. To save weight and space brass bearings were used. Unfortunately, the higher friction produced by the brass bearings in combination with an under-powered (but light weight), z-axis drive motor did not work well. Motion was intermittent and the motor often stalled. With the test experience we now have, we believe the X and Y drives have sufficient power to permit the use of heavier linear bearings and a larger z-axis drive for any follow-on units we might build.

System testing with high accelerations showed undesirable X-axis deflections in the manipulator frame caused by high acceleration moves in the X direction. To cure this problem, we designed a set of dampers for the manipulator frame to reduce the vibration. These dampers are currently under construction and will be installed as soon as construction is complete.

4.1.3 Payload design

We define the payload as including the touchboard (the panel on which all TOPIT simulated cockpit controls are mounted) and all associated servos, solenoids, and other hardware. It is the payload which is moved by the manipulator.

Rotary and toggle switches on the payload had to be rotated so that the user would find the switch in the correct position corresponding to the image of the virtual switch in his HMD. A number of alternative configurations of gears and motors that could accomplish the needed rotation were considered. The simplest way would have been to directly rotate each switch, without gearing, [Figure 4.1-12]. In the figure solenoids are used to engage switch détentes under computer control so that switches having different détente spacing could be simulated.

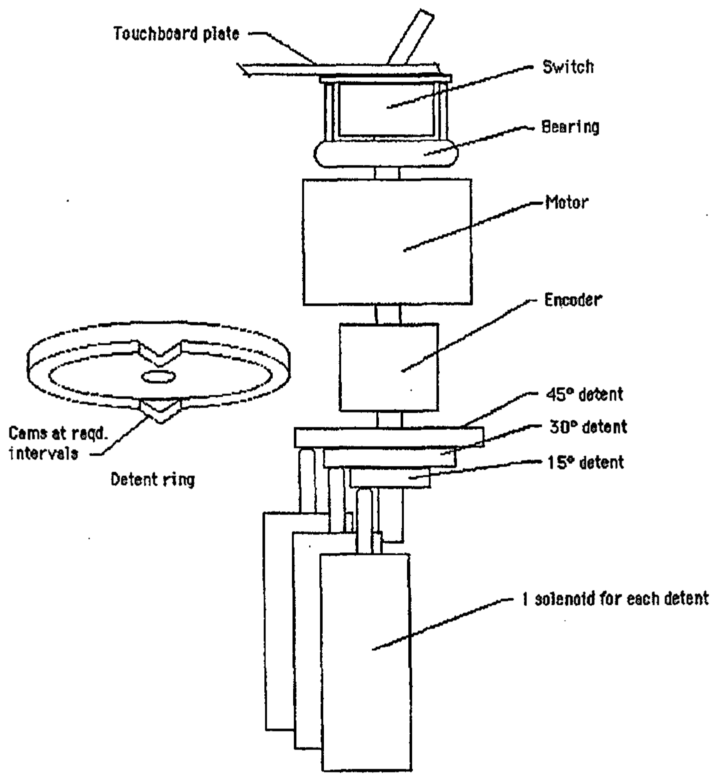


Figure 4.1-12 Direct drive switch rotation with programmable détente positions.

Working from the sizes and weights of the various switches and the properties of motors, the design implications of the alternatives were studied. Table 4.1-1 shows an example of the calculations we did for a particular motor and gear configuration.

Table 4.1-1 Payload design calculations.

Switch	Average Switch Speed (rad/sec)	Switch acceleration (rad/sec²)	Average motor speed (rad/sec)	Motor acceleration (rad/sec²)	Armature torque (oz. in..)	Friction torque (oz.in.)	Load torque (oz.in.)	Motor torque required (oz.in.)
A1	7,200	10,048	7,200	10,048	1.68	0.17	3.25	5.10
A2	7,200	10,048	7,200	10,048	1.68	0.17	2.75	4.59
A3	7,200	10,048	7,200	10,048	1.68	0.17	0.55	2.40
A4	7,200	10,048	7,200	10,048	1.68	0.17	0.37	2.21
B1	5,400	7,536	5,400	7,536	1.96	0.28	7.63	9.87
B2	5,400	7,536	5,400	7,536	1.96	0.28	7.63	9.87
B3	5,400	7,536	5,400	7,536	1.96	0.28	7.63	9.87
B4	5,400	7,536	5,400	7,536	1.96	0.28	7.63	9.87

The rotary switch portion of the payload is a mechanism that moves selector switches and continuous rotary controls (like volume controls) into the angular positions to which they were last set by the user. The user must find each control in the correct angle and with the correct détentes and stops for that control. A motor turns the control to the correct position and a set of solenoids selects the détentes. A second motor (not shown) selects the stop angles for the selected rotary control. The stop motor controls the extreme left and right control rotation angles.

In many ways the rotary control portion of the payload design is the most demanding, because a complex mechanism must be put in a small space. To save weight, a single motor and solenoid mechanism was geared to drive four rotary controls. Only one of the four rotary controls can be accessed at any one time by the user, so it makes no difference that the other knobs linked by gears happen to be rotating in unison.

4.1.4 Control software development

At the beginning of the project, we shared some of the assets of this contract with the *STRICOM SBIR A94-062 3-Axis Locomotion Simulator Study* contract for basic motion control software development efforts. Basic motion control is common to both projects. The servo electronics were connected to a servo motor we installed in a commercially available treadmill. The treadmill was used to demonstrate single axis control for the *SBIR A94-062* project. Control software developed for the treadmill demonstrator was modified for use on the TOPIT.

Treadmill demonstrator control software accepted tracker position data which indicated the treadmill user's position on the treadmill and adjusted the speed of the treadmill to prevent the user from walking or running off either end of the treadmill. User tracking was first done mechanically with a "stick" tracker - a piece of wood with one end attached to an encoder and the other end held by the treadmill user. The stick tracker was sufficient for its purpose and served us well during our initial motion control experiments.

Efforts then progressed to the Ascension magnetic tracker later in the effort. Software which phase locks the PMAC operations to those of the PC were also developed and proven with the treadmill demonstrator. This phase locking software was directly applicable to the TOPIT.

Safety software was added limit the speed and acceleration of the servo motor. The safety software serves two purposes. It limits the speeds and accelerations to avoid damaging the servo and drive mechanism, and it protects the user from high accelerations that might cause a loss of balance. Initially, the limits were set low to protect the user. As we got the tracking and control algorithms perfected, the envelope of the treadmill performance was expanded to accommodate more vigorous acceleration, running, and stopping.

Initial motion control testing on the desktop prototype was done with user input directly to the PMAC via an encoder [Figure 4.1-13] . The encoder (upper left in the figure) mounted temporarily in the electronics cabinet was used to provide a temporary, electrically noise-free source of hand position tracking. Next we drove an analog input to the PC with a potentiometer to check the PC/PMAC interface. We then drove the system with the magnetic tracker input to the PC. Subsequent testing used the Multi-Sensor Hybrid Tracker (discussed below). This gave us the ability to move the tracker and have the desktop demonstrator payload follow. This step-by-step checkout procedure allowed us to thoroughly check each component before adding another uncertainty to the system.

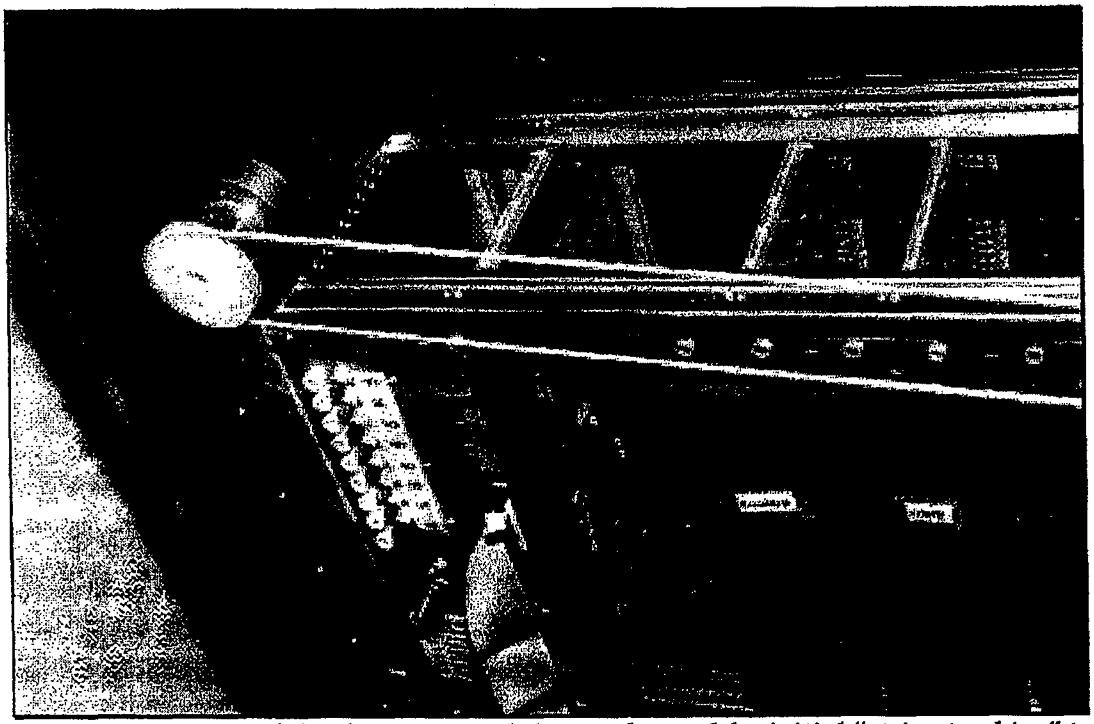


Figure 4.1-13 Portion of the electronics and the encoder used for initial "string tracking" tests - the string loop extends to the prototype fixture and slider motion tracks the string position.

To get best performance from the system it was necessary to tune motor performance to the actual masses and spring constants of the respective motor loads. Delta Tau, manufacturer of the PMAC motion control card we are using, provides a set of software utilities for this purpose.

4.2 Tracking System

Project Objective #2: ***Ensure the tracking system provided sufficient accuracy in the presence of electromagnetic noise and moving metal objects.***

Our hybrid tracker consists of a commercially available Ascension Flock of Birds magnetic tracker combined with a custom six degree-of-freedom inertial tracker. The magnetic tracker produces both position and attitude data but, to minimize inherent noise, data averaging of many sequential data points is used. This averaging process introduces a time lag in the position and attitude data sent to the computer. A second problem with magnetic trackers is their susceptibility to interference by metallic objects - particularly those containing aluminum, steel, or iron.

To solve the lag problem we constructed a hybrid tracker by adding inertial tracking to the magnetic tracker. The inertial tracker measures X, Y, and Z linear accelerations as well as rotational accelerations in roll, pitch, and yaw. Kalman filter software running in a Pentium Pro 200 is used to combine the data from the magnetic and inertial sensors to provide accurate, low lag hand position and attitude information for the simulation.

4.2.1 Hybrid tracker development

We ran some preliminary experiments to determine the tracker's susceptibility to interference from non-moving steel and aluminum objects. The tracker was also placed near a 2 horsepower electric motor. The motor was turned on and off to roughly simulate what might happen with a servo. The results showed about what we expected: the tracker is quite susceptible to the motor noise. We also ran some additional experiments to determine the susceptibility to interference from non-moving carbon steel, stainless steel, and aluminum objects which were placed nearby. There were several interesting results. First, the tracker exhibits a large error at longer tracker ranges (24 inches or more) and the error is not affected much by nearby non-moving interfering test objects. We also noted that stainless steel test objects had no noticeable affect on the tracker accuracy. We also determined that the shape of aluminum test objects had a large effect on the tracker accuracy.

A detailed list of tests and software required to test the magnetic tracker was prepared. The purpose of those tests was to determine the effect of stainless steel, carbon steel, and aluminum on the accuracy of the tracked position. For this series of tests, test objects were placed at various distances from the tracker receiver and the tracker receiver will be placed at various distances from the tracker transmitter. Test objects included one inch round by two foot long tubes and one foot by two foot sheets of the three metals mentioned above. Test results showed no significant interference with series 300 stainless steel while also showing there was substantial interference with both carbon steel and aluminum and thus confirmed what we had been told by the magnetic tracker manufacturer Ascension.

We looked at ways to overcome the noise susceptibility of the magnetic tracker, and developed an approach which used inertial sensors in conjunction with the magnetic tracker. Inertial sensors, miniature accelerometers and angular rate sensors, provide excellent short-term accuracy that is immune to electromagnetic effects. However, the inertial measurements drift over time, and must be updated with position and angle "fixes" to remove the drift errors.

If magnetic tracker measurements are available occasionally to update the inertial measurements, then we expected the tracking system would perform well over both the short term and long term.

Kalman Filter Software

The Kalman filter software combines the data from all the sensors under consideration to provide a combined "optimal" estimates of position and attitude. The Kalman filter software actually combines data from various sensors continually, weighting the value of each data according to its error characteristics. It works amazingly well. Kalman filter technology has been used in aerospace applications such as aircraft navigation for a long time.

Initially, most of the sensor error models were based on published specifications, but as we collected data the error models were improved.

Kalman filter software requires "tuning" as part of the test process before it will function properly. Tuning is the adjustment of the mathematical models of the sensor error performance to agree with the error performance encountered in practice. Paul worked with one of our student interns to tune the software.

In the development we captured a set of hybrid tracker data for a small set of hand motions: left - right, fore - aft, and a series of rolls. The data were used to tune the Kalman filter software. Rather than sampling data at different times, all of our data is sampled at one instant in time. Changes were made to the software, which was originally designed to sample data at different times, to accommodate the new data sampling scheme and forwarded revised software to CGSD.

Magnetic field calibration software

Despite the use of series 300 stainless steel for most custom portions of the TOPIT the magnetic field of the tracker was still distorted by the presence of solenoids, motor, switches and other non-stainless steel metallic items. To get the needed accuracy from the magnetic tracker in the presence of these items we developed software to map the position dependent magnetic field distortions for the magnetic tracker. The magnetic tracker was attached to the payload. The payload was driven to various positions by custom calibration software. The position and attitude at each grid location was recorded in a table. The contents of the magnetic calibration table were subsequently used by the real time software to determine the actual location of the magnetic tracker.

TEU Hardware

At the time we started development of the hybrid tracker we had three contracts which required advanced tracker technology; this contract, the *STRICOM SBIR A94-062 3-Axis Locomotion Simulator Study* contract, and a commercial contract. We constructed a tracker evaluation unit (TEU) which includes X, Y, and Z accelerometers, roll, pitch, and yaw angular rate sensors, a tilt sensor, and a compass. This module along with custom software was used to select the sensor configuration most appropriate for each of the three applications. A printed circuit board was designed, fabricated, assembled [Figure 4.2-1], and tested. As described above, the TEU has a large collection of sensors and is designed strictly to support lab experimentation. It is too large and has too many sensors for production. However, it is essential for collecting the real data we need to support algorithm development.

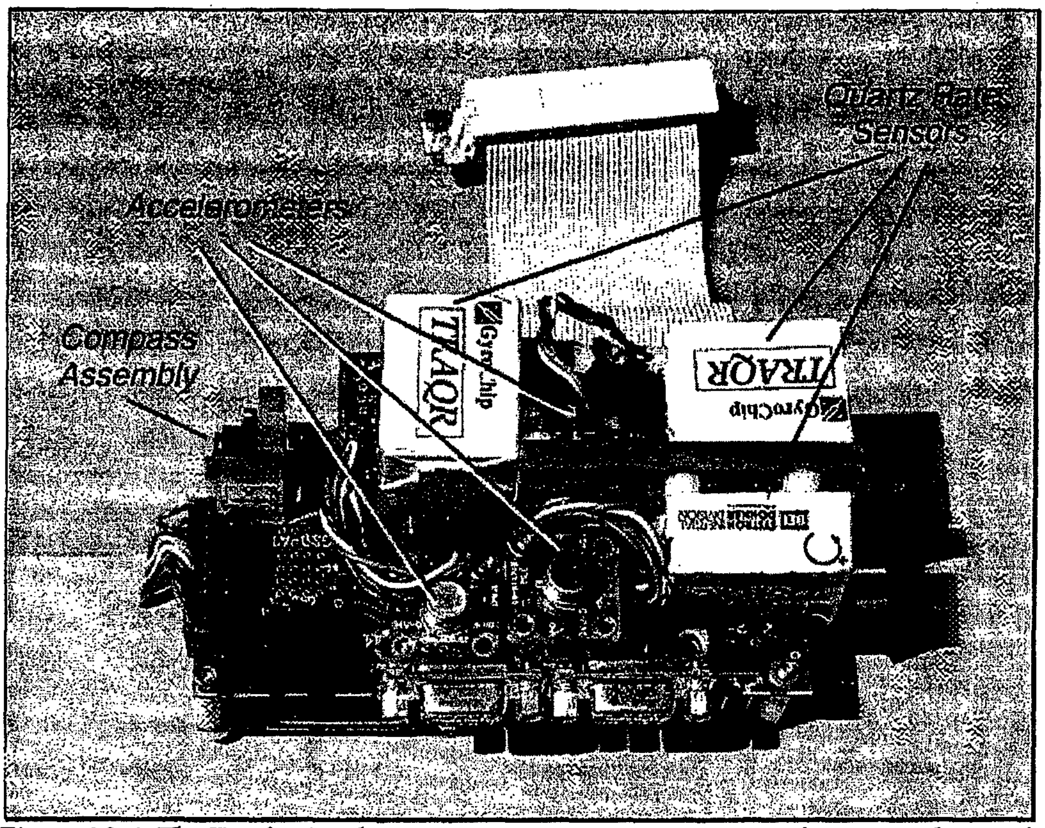


Figure 4.2-1 The Tracker Evaluation Unit contains more sensors than were ultimately selected.

Hybrid Tracker 2 Hardware

We decided to repackage the hybrid tracker so that it would not be as bulky and heavy. The original TEU discussed above contains a compass and tilt sensor which are not required by the TOPIT program. The redesigned unit, called "hybrid tracker 2", consists of two modules; a sensor module shown with the magnetic tracker [Figure 4.2-2] which contains the gyros and accelerometers and the control module for the computer interface. The custom printed circuit board, compass, and tilt sensors used in the TEU were eliminated. The sensor module, which is attached to the back of the glove, is much smaller and lighter than TEU. It is connected to the control module by a single cable.

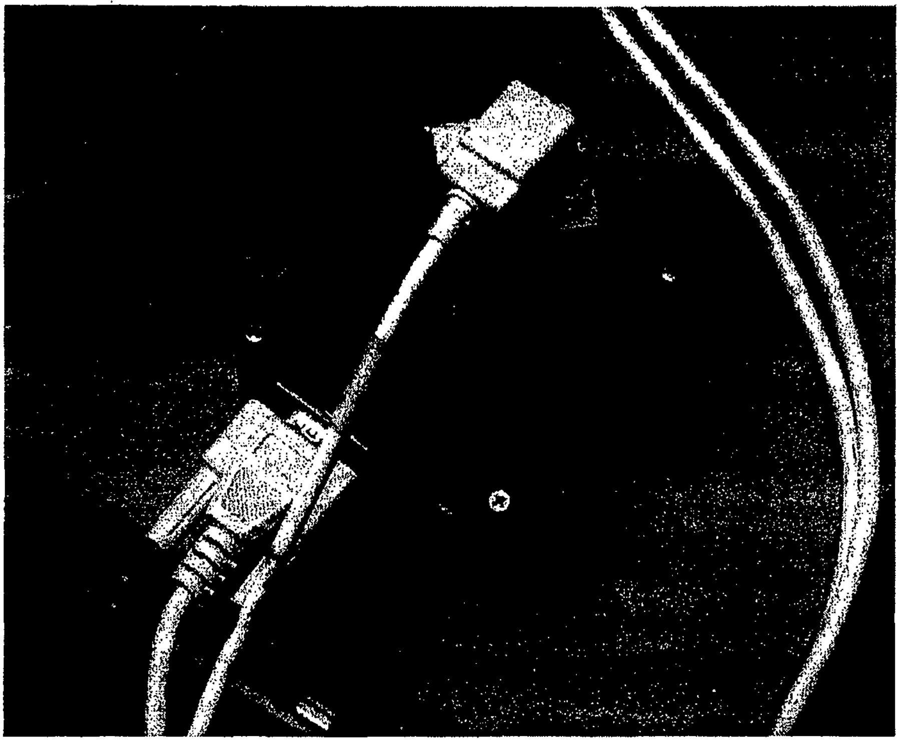


Figure 4.2-2 Hybrid tracker 2 sensor module with magnetic tracker attached.

We believe a production version of the hybrid tracker can be considerably smaller and lighter weight than hybrid tracker 2 shown.

Hybrid Tracker 3 Hardware

Our research showed that the alignment of the sensors in the Hybrid Tracker 2 unit was not acceptable. As a result we integrated two triaxial sensors - a rate gyro sensor unit and an accelerometer sensor unit. This replaced the inertial tracking portion of the Hybrid Tracker 2 unit. We are still using the Flock of Birds magnetic tracker unit. Hybrid Tracker 3 is lighter and has a smaller footprint than its precursors. Signal conditioning hardware, built by one of our student interns, was used to integrate the new sensors with the existing system.

4.2.2 Optical Tracker Development

Since the result of the hybrid tracker is not quite up to our expectations, we are further exploring the optical tracker.

We have installed DynaSight™ sensor (optical tracker by Origin), have written drivers for the optical tracker, and integrated the optical tracker into TOPIT.

Two methods were integrated in order to test the applicability of each. The two methods are described below:

Method 1:

Method one uses a single optical target. The target used was initially just a piece of retro-reflective tape applied to the glove's index finger. Offsets were recalculated for this tracking system and the hand position was backwards calculated from the glove offsets in order to simulate the finger articulations. The hand orientation was read in from the magnetic tracker.

This proved to get rid of some of the position noise as well as problems with the warped magnetic field (caused by the stainless steel frame and the EM fields from the motors and solenoids in the payload); however, after testing, a position hysteresis was found while moving along the X axis of the optical tracker. A constant offset was noticed with constant velocity. This was a result of the sequential (180 degrees our of phase) measurements taken by each optical sensor as it calculated the Z axis.

The retro-reflective tape was exchanged for an active target (IR LED). Integrating the IR LED, the ATA (Active Target Adapter), and the TOPIT™ system was easily accomplished. DIP switches in the back side of the optical tracking unit can be set to use the ATA and IR LED (please refer to the DynaSight™ Sensor manuals).

Method 2:

Method two uses three optical, active targets in order to calculate the hand orientation as well as position. This required writing a driver for the QNX operating system. The ATA integrated easily with the DynaSight™ Sensor (DIP switches must be adjusted. Please refer to the DynaSight™ Sensor manuals). The three active targets were mounted on a triangular plate (supplied by Origin Instruments) and mounted on the CyberGlove.

The purpose of using optical sensors was to increase the position accuracy of the sensors. Without filtering, the position measurements were much less noisy compared to the magnetic trackers. Also, since the optical trackers are not affected by EM waves, the warped field did not affect its accuracy.

Notes in comparing the noise to that of the magnetic tracking unit (MT) are as follows:

Noise reading: (taken on 26 February 1998)
- MT with ALL filters on (lots of lag – unacceptable lag)
  - position: 0.02 inches
  - angle: 0.03 degrees
- MT with ACNarrow filter only (normal operation mode)
  - lag is okay...
  - position: 0.85 inches
  - angle: 1.70 degrees
- OPTICAL (no filtering)
  - lag is okay...
  - position: 0.06 inches
  - angle: 1.45 degrees

Because the magnetic tracker measurements with all of the filters on creates an unacceptable lag, these noise figures are not used as comparison. With the ACNarrow filters on only, the magnetic tracker data contains more noise compared to that of the optical tracker.

Method 1 with an active target (IR LED) is used even though its implementation does NOT output angle information. The reason for this is that the position of the fingertip is given rather than the position of the wrist (as the case would be in Method 2). If the sensor is used to calculate the wrist position, the angle noise would affect the calculated fingertip position. Therefore, Method 1 provides a more accurate solution to the fingertip tracking problem.

4.3 Hand Motion Prediction Algorithms

Project Objective #3: ***Design hand motion prediction algorithms that predict which control will be touched while sufficient time remains to put it in place.***

Software to meet this objective consists of two parts; the hand motion prediction algorithm itself, and the cockpit to real time capture zone software.

Hand motion prediction

The hand motion prediction algorithm was implemented by starting with current hand position and attitude. Predicted positions and attitude is determined by using accelerations and velocities for six degrees of freedom. The general form of the computation, which is performed in all six axes of motion (X, Y, Z, roll, pitch, and yaw), is as shown in equation 1. The hand prediction algorithm is working well.

$$P_P = P_C + V_C T + A_C T^2 \qquad \textit{[Eqn. 1]}$$

where:

P_P = predicted position

P_C = current position

V_C = current velocity

A_C = current acceleration

T = time to predicted position

Zone capture software

It was useful to divide the operating envelope of the virtual cockpit into regions or "zones" that help define TOPIT FTFS manipulator operating conditions and operating actions. The four zones are:

Zone A: A volume ordinarily containing the user, when the user is not accessing any controls on the virtual control panel.

Zone B: A volume outside Zone A extending to within a short distance - about 1.5 inches - of the surfaces of the front of the virtual control panel with its instruments.

Zone C: The volume on the user side of the virtual instruments within a short distance - about 1.5 inches - of the virtual control panel with its instruments but outside Zone B.

Zone D: The volume beyond Zone C, including the surfaces of the controls and the control panel.

Figure 4.3-1 illustrates a cross-section with the various manipulator control zones.

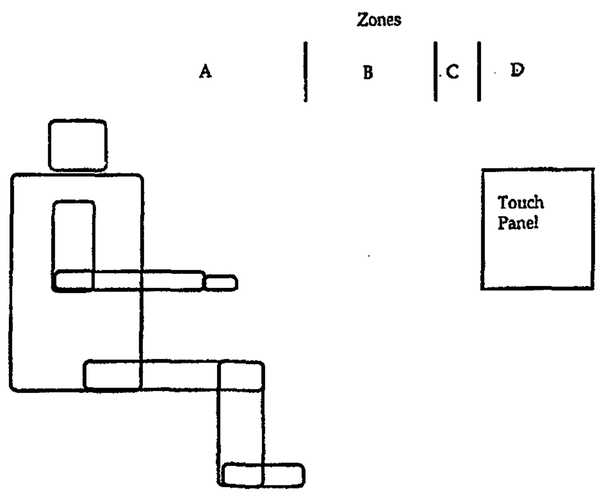


Figure 4.3-1 Manipulator Control Zones.

Note that for the prototype TOPIT, the controls need not lie in a plane. The transport includes a depth axis which allows the virtual control panel to be stepped or even curved. Accordingly, the control zone volumes A-D are mapped to the contours of the cockpit control panel in a lookup table and are not bounded by planes.

The requirements for TOPIT FTFS positioning speed depend upon the position of the user's hand relative to the operating zones described above. Motion requirements for each operating zone follow:

Zone A: When the user's tracked hand is within Zone A, the TOPIT FTFS will not move.

Zone B: When the user's tracked hand is in Zone B, the TOPIT FTFS will move at maximum speed to the mirror point. The mirror point is that manipulator position which is closest to the user's tracked hand.

Zone C: When the user's tracked hand is in Zone C, the TOPIT FTFS manipulator will move at maximum speed. In this zone the computer determines which control the user is reaching for and positions the appropriate control at the correct position in front of the user. The choice made by the computer will be based upon extrapolation of the hand trajectory. The final position of the TOPIT FTFS is not made until the user's hand enters zone D.

Zone D: When the user's tracked hand is in Zone D, the TOPIT FTFS will not move.

The real time capture zone software that provides these functions operates properly.

Early in debug we discovered the need for an additional refinement. As originally implemented, the control software would always attempt to use the maximum performance of the hardware to track the hand. It would accelerate the payload at 4 Gs to follow the hand motion, even when the hand was relatively far away from the simulated panel surface in zone B. The resulting violent motion of the hardware place unnecessary stress on the mechanism, because high accelerations are only required when the users hand is relatively near the panel.

To damp the motion of the payload when the hand is distant, we implemented a variable time constant filter that smoothes the motion tracking. When the hand is far away, the time constant is large and high accelerations are avoided. The time constants are reduced to provide full performance as the hand approaches a control on the panel.

Note that the additional smoothing is not applied to the position used for rendering the image of the hand. The image always react quickly so the user will have a correct view of his hand position. The extra filtering is not applied to the extrapolated hand position used to select which control will be actuated. The extra filtering is only applied to the payload positioning commands.

4.4 Computation Control Lags

Project Objective #4: ***Keep computation and control lags small enough so that the positioning system had sufficient time to position the touchboard.***

The following steps were taken to minimize computation and control lags in the system:

- Dedicated servo controllers were used to provide high computational rates to support the servo loop computations. The loop update rates are over 100Hz.
- True real time operating systems were used throughout: Ultrix, SGI's real time version of UNIX in the Onyx; QNX, a proprietary real time operating system, in the PC; and the PMAC motion control system in the servo controllers. True real time systems allow the user to manage the priorities of interrupts so that time critical events are not delayed in a service queue.
- A dedicated Ethernet™ link was used between the PC and the Onyx. Having a dedicated link avoids latency due to packet collisions, and latency is kept under one millisecond.
- The hybrid tracker provides accurate rate and acceleration data to extrapolate over unavoidable latencies, such as the time in the image generator needed to render the graphics imagery.

Overall, system latencies are quite good, but there are two limitations. First, we could not afford to build a second hybrid tracker for the head mounted display, so there are noticeable lags when head motion is rapid. There is no technical problem in adding a second hybrid tracker. Second, even though a substantial amount of the budget was devoted to obtaining a high performance image generator, and considerable effort was made to minimize polygon counts in the database, the image generator frame rate is at most 30Hz, and it sometimes drops as low as 20Hz. Image generator technology has advanced rapidly, so that now at the end of the two year development program there are available image generators with much higher polygon capacities than the SGI RealityEngine2. For example, the newer Lockheed-Martin Real3D Pro 2000 provides about three times the polygon capacity at one-third the cost of the RealityEngine2.

4.5 Safety Systems

Project Objective #5: ***Provide redundant safety systems to protect the operator during development and use.***

A number of things were done to ensure personnel and equipment safety. Some of these features were discussed briefly in previous sections. These design features, combined with a few common sense procedures have served us well - we have had no injuries on the TOPIT program. The design features and procedures are discussed below.

To ensure the user's head is not hit by the moving payload, should he lean forward for some reason, the user's seat is positioned as far away from the manipulator frame as practical. The seat position still allows the user to activate touchboard controls without excessive leaning.

To ensure the user's untracked left hand is never near the moving mechanism, the user must depress a button on the throttle with his left thumb. If he lets go of the button, real-time software causes X and Y motions to stop (other motions are considered not dangerous). The system must be reinitialized by the system operator before it will move again.

Limit switches on the X, Y and Z axes are used for system initialization [Figure 4.5-1]. Limit switches are located near the ends of the excursions on each axis and are activated if the payload touches them. Limit switches should not be activated during real time simulation, i.e., a motion control error has occurred if they do. If any of these switches are activated during real time simulation, the PMAC motion controller software causes all motions to stop. The system must be reinitialized by the system operator before it will move again.

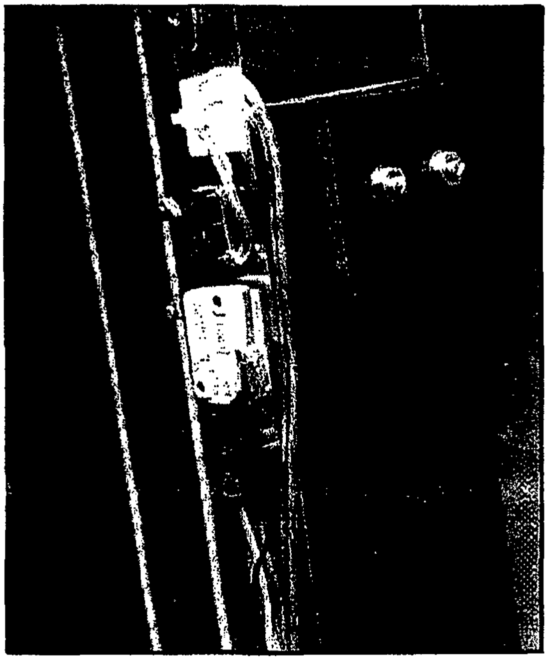


Figure 4.5-1 Typical limit and emergency limit switches.

A set of emergency limit switches is located next to but beyond the X and Y limit switches [Figure 4.5-1]. Should the limit switches or the PMAC software discussed above fail to act for any reason, the emergency limit switches will cause the X and Y drive motor windings to be disconnected from their respective amplifiers and shorted together. This will cause motion in both the X and Y axes to stop immediately. Two manually operated emergency stop switches, one located on the electronics cabinet and one near the user's left hand, can also be used to stop X and Y motion. Manual reset is necessary to reactive the system following such a shut down.

A potential hazard on any mechanical design is sharp edges. These were minimized during the design process. With a moving mechanism, such as the TOPIT, the concern becomes the existence and control of pinch points. We use both bumpers and shields (not yet installed) to minimize personnel exposure to pinch points.

To minimize exposure of the user's tracked right hand to contact with moving parts of the TOPIT we programmed the hand motion prediction algorithms to stop X, Y, and Z motions and freeze the position of the payload when the hand approaches the payload. Payload motion does not resume until the user moves his hand away from the payload again.

A safety light flashes when manipulator power is on. The safety light is positioned so that it can be seen by anyone in the area of the TOPIT (except the user when he dons the HMD).

Turning to procedures, we have a two man rule for TOPIT operation when the user is wearing the HMD and cannot see the manipulator. The second man positions himself close to one of the manually operated emergency stop switches so that he can activate the switch should he see anything out of the ordinary.

To prevent a passer-by from being injured, should a computer glitch cause unexpected manipulator motion, we insist TOPIT personnel shut off manipulator power when they are not in the vicinity of the machine.

5. Conclusions

Overall, the major technical challenges were met. In particular, robotic hardware was built to position the controls with the speed and accuracy required, and a sophisticated tracker and an alternative tracker were built to provide the accuracies required for position and extrapolation. The most difficult aspect of the program turned out to be getting all of the bugs out of the complex system under severe budget constraints. In this last respect we were largely successful, but not entirely. The main limitations of the final prototype lie in the fine points of getting the software to run completely smoothly and reliably. We view none of the present limitation as being fundamental.

THIS PAGE INTENTIONALLY LEFT BLANK

Electronic References

THIS PAGE INTENTIONALLY LEFT BLANK

Internet Locations

Note: The following URLs are current as of the date of publication.

Indiana University; Analog VLSI & Robotics Laboratory
www.cs.indiana.edu/robotics/world.html
This site provides links to a multitude of other cybernetic-related web sites.

MIT – Artificial Intelligence Laboratory
www.ai.mit.edu/publications/pubsDB/pubsDB/onlinehtml
This site contains downloadable documents. Newer research is toward the bottom.

American Association for Artificial Intelligence
www.aaai.org/

The Electronic Journal of Haptics Research
www.haptics-e.org/
Contains current research papers regarding innovations in the field of haptics.

CRDI of Robotics and Technical Cybernetics
www.rtc.neva.ru/english/index1.html

Space Automation and Robotics Technical Committee
www.ssl.umd.edu/SARTC_html/SARTChome.html

University of CA – Berkeley; Robotics and Intelligent Machines Laboratory
robotics.eecs.berkeley.edu/

Robotics at Space and Naval Warfare Systems Center, San Diego
www.nosc.mil/robots/

Department of Defense, Joint Robotics Program
www.jointrobotics.com/

Robocup
www.robocup.org/
This is the homepage for an annual cybernetics competition. Long term goals in the realm of cybernetics are explored.

The Robotics Institute
www.ri.cmu.edu/cgi-bin/tech_reports.cgi
This site links users to a number of downloadable robotics and cybernetics research papers.

The Haptics Community Web Page
www.haptic.mech.northwestern.edu/

Navy Center for Applied Research in Artificial Intelligence
www.aic.nrl.navy.mil/projects.html

DTIC-Matris – Los Angeles
dticam.dtic.mil/hm/org.html

Additional References

Distribution A – Approved for public release; Distribution is unlimited.

Note: Refer to the order form following the bibliography for ordering information.

THIS PAGE INTENTIONALLY LEFT BLANK

♦ **AD**-A382305

ARIZONA UNIV TUCSON DEPT OF ELECTRICAL AND COMPUTER ENGINEERING

Soldier Performance Course of Action (COA) Visualization Aids

01 Aug 2000 33 PAGES

PERSONAL AUTHORS: Rozenblit, Jerzy W.; Barnes, Michael J.; Momen, Faisal; Quijada, Jose A.; Fichtl, Theodore

ABSTRACT: The computer revolution has resulted in extending the possibilities of battle space visualization to the brigade commander and below. However, mobility and bandwidth considerations require that the systems be efficient to reflect the realities of modern combat. The Advanced Battlespace Architecture for Tactical Information Selection (ABATIS) is being developed to be a rapid planting and re-planing experimental environment. ABATIS's object-oriented architecture has the advantage of being able to rapidly construct a three-dimensional battle space that will accurately represent the essential planning components of a brigade and smaller division battle environment. The basic architecture has been extended to include war-gaming logic as part of the software design, and examples are given that pertain to specific military problems. This capability will allow ABATIS to realize fully the implications of battle space visualization by creating a human computer synergy that encourages both human and machine to generate and evaluate possible courses of action and their consequences. The human performance implications are discussed, and particular attention is directed toward research issues related to terrain visualization, automation, decision making, and cognitive biases.

DESCRIPTORS: *ARMY TRAINING, *DECISION AIDS, *VIRTUAL REALITY, *ABATIS(ADVANCED BATTLESPACE ARCHITECTURE FOR TACTICAL INFORMATION SELECTION), AUTOMATION, STRATEGIC ANALYSIS, COMPUTER ARCHITECTURE, PERFORMANCE(HUMAN), WAR GAMES, MAN COMPUTER INTERFACE, ARMY OPERATIONS, TACTICAL DATA SYSTEMS, VISUAL AIDS, TACTICAL INTELLIGENCE, DECISION SUPPORT SYSTEMS.

AD-A380074

ARMY WAR COLL CARLISLE BARRACKS PA

Integrating Knowledge Management Initiatives for the Future Army

05 May 2000 80 PAGES

PERSONAL AUTHORS: Castro, Felix D., Jr

ABSTRACT: Using the Wisdom Triangle/Pyramid as its foundation, this research paper provides recommendations for an Army Knowledge Strategy, which combines the value of sharing business knowledge with today's computer industry's pursuit at developing a "human machine." One of the best business practices is sharing knowledge within the company/firm. If the Future Army is to maintain its reputation as being the most potent land combat force in the world, it must broaden its efforts to identify, capture, prioritize, organize, create and share Army knowledge. Innovations of today's computer industry advocate the development of a "human machine," a machine that thinks and learns on its own, a machine that imitates a man's brain. The Army should leap beyond the mere exploitation of today's technology and keep abreast with this future innovation. Tomorrow's technology will be the conduit for sharing knowledge not only among soldiers and its future generations, but also among unmanned weapon platforms integrated together to share knowledge (of the battle situation, terrain, courses of action, and threat information) in achieving a commander's objective on the battlefield or to secure national security objectives. The Army should begin implementing initiatives involving data, information and knowledge that provide the basis for machines to think and learn, as unmanned weapon platforms will one day be a reality. Incorporating the recommendations as explained in this paper offers a simple strategy from which to build a Knowledgeable Army.

DESCRIPTORS: *MANAGEMENT INFORMATION SYSTEMS, *ARMY PERSONNEL, *ARTIFICIAL INTELLIGENCE, *MAN MACHINE SYSTEMS, *CORPORATE INFORMATION MANAGEMENT, ARMY PLANNING, INFRASTRUCTURE, WISDOM TRIANGLE, WISDOM PYRAMID, WISDOM TRIANGLE/PYRAMID.

♦ Included in the DTIC Review, March 2001

AD-A380056

AIR FORCE RESEARCH LAB WRIGHT-PATTERSON AFB OH HUMAN EFFECTIVENESS DIRECTORATE

Brain-Computer Interfaces Based on the Steady-State Visual Evoked Response

01 Jun 2000 13 PAGES

PERSONAL AUTHORS: Middendorf, Matthew; McMillan, Grant; Calhoun, Gloria; Jones, Keith S.

ABSTRACT: The Air Force Research Laboratory has implemented and evaluated two Brain-Computer Interfaces (BCI's) that translate the steady-state visual evoked response into a control signal for operating a physical device or computer program. In one approach, operators self-regulate the brain response; the other approach uses multiple evoked responses.

DESCRIPTORS: *BRAIN, *MAN COMPUTER INTERFACE, REPRINTS, NEURAL NETS, AIR FORCE RESEARCH, RESPONSE(BIOLOGY), ELECTROENCEPHALOGRAPHY, VISUAL SIGNALS, SELF ORGANIZING SYSTEMS, BIOFEEDBACK, BCI(BRAIN COMPUTER INTERFACE), SSVER(STEADY STATE VISUAL EVOKED RESPONSE).

♦ AD-A378892

SYTRONICS INC DAYTON OH

Development Manual for 3D World Virtual Environment Software

01 Dec 1999 59 PAGES

PERSONAL AUTHORS: McCoy, Annette L.; Schnipke, Susan K.

ABSTRACT: This report documents a software package called 3D World. The software provides the environment and scenario development tools necessary to create a virtual environment for human performance research. This report contains step-by-step instructions on how to develop and run virtual environments, as well as an in-depth description of the program structure.

DESCRIPTORS: *VIRTUAL REALITY, *SOFTWARE TOOLS, DATA BASES, COMPUTER PROGRAM DOCUMENTATION, IMAGE PROCESSING, SCENARIOS, PERFORMANCE(HUMAN), THREE DIMENSIONAL, COMPUTER GRAPHICS, 3D WORLD COMPUTER PROGRAM, VIRTUAL ENVIRONMENTS.

♦ Included in the DTIC Review, March 2001

AD-A374699

SCIENCE APPLICATIONS INTERNATIONAL CORPDAYTON OH

Imagery Analyst Workstation User-Interface Analysis and User-Interface Requirements Document

01 Dec 1999 60 PAGES

PERSONAL AUTHORS: Malek, David A.; Brett, Bryan E.; Martin, Edward A.; Hoagland, David G.

ABSTRACT: This document describes the approach and method employed in conducting an Imagery Analyst and Intelligence Analyst Workstation Interface analysis and documents the results and recommendations from this analysis including a user interface requirement document for imagery analysis workstations.

DESCRIPTORS: *WORK STATIONS, *IMAGE DISSECTION, *GRAPHICAL USER INTERFACE, HUMAN FACTORS ENGINEERING, IMAGE INTENSIFICATION, USER NEEDS, IMAGE RESTORATION, FEATURE EXTRACTION.

AD-A373202

SOUTH CAROLINA UNIV COLUMBIA

Contributions to the Development of VE-Assisted Training of Spatial Behavior

01 Feb 2000 24 PAGES

PERSONAL AUTHORS: Allen, Gary L.

ABSTRACT: A conceptual analysis for identifying essential spatial skills concluded that three functional families of skills can be identified (object identification, wayfinding and orientation, and target interception/interception avoidance) and that current VE technology is sufficient for their assessment and training. An empirical study of frame of reference control indicated that global and local object frames of reference were more effective than mobile object based and observer based frames in fixed observer situations. An empirical study of distance estimation training showed that skill in estimating metric distance can be trained in the field using either verbal or visual feedback and transferred to a novel field setting.

DESCRIPTORS: *SPATIAL DISTRIBUTION, *PERCEPTION(PSYCHOLOGY), *VIRTUAL REALITY, PERFORMANCE(HUMAN), COGNITION, VISUAL PERCEPTION, NAVAL TRAINING, CONDITIONING(LEARNING), TRANSFER OF TRAINING.

AD-A373117

DEFENSE ADVANCED RESEARCH PROJECTS AGENCY ARLINGTON VA

Advances in Enterprise Control. AEC Proceedings, November 15-16, 1999/San Diego, California

16 Nov 1999 299 PAGES

ABSTRACT: This document contains copies of 33 papers prepared for and presented at the November 1999 DARPA JFACC Symposium on Advances in Enterprise Control. The purpose of the Symposium was to bring together researchers and practitioners from industry, government and academia to present and discuss the latest developments in all aspects of enterprise control. The Symposium presented papers that (a) describe the results of original research on the topics of interest, (b) provide broad reviews of the state-of-the-art, and(c) propose and advocate new research directions. Also presented were papers that describe significant practical experiences with current enterprise control systems: complex dynamic phenomena, non-obvious successes and failures, requirements and unmet needs. The modern enterprise is a large-scale dynamic system with broadly distributed and potentially conflicting goals, resources and constraints, with multiple semi-autonomous participants of both human and artificial nature (e.g., large military operations, financial/trading institutions, logistics systems, manufacturing plants, power grids). The increasing capabilities of technology to collect, automatically generate, and disseminate information offer the possibility for large-scale enterprises to be more responsive to change. Enterprise plans and orders quickly become obsolete as new information about the current situation becomes available. The challenge is to use real-time information to re-direct enterprise operations effectively. Such systems and challenges define the scope of the Symposium.

DESCRIPTORS: *HUMAN FACTORS ENGINEERING, *ARTIFICIAL INTELLIGENCE, *GAME THEORY, SYMPOSIA, ADAPTIVE CONTROL SYSTEMS, DISTRIBUTED DATA PROCESSING, REAL TIME, COMMAND AND CONTROL SYSTEMS, CONTROL THEORY, DYNAMIC PROGRAMMING, DECISION SUPPORT SYSTEMS.

♦AD-A373078

STOTTLER HENKE ASSOCIATES INC SAN MATEO CA

An Intelligent Training Management System (ITMS)

28 Jan 2000 83 PAGES

PERSONAL AUTHORS: Stottler, Richard

ABSTRACT: Air Force training units are in extreme need of advanced, intelligent training management systems to aid the training managers and schedulers in the performance of their duties and to help students quickly advance in their careers and meet the training requirements. The Intelligent Training Management System (ITMS), to be implemented and used in Phase 2, will address tracking, evaluation, requirements identification, scheduling, and completion and certification management of individuals and teams. The ITMS will perform the functions that a person dedicated to managing the training of a small group of students would perform, but do it automatically. It will intelligently guide the students as to their training needs and opportunities and help with the development, delivery, scheduling, and evaluation of courses and other training events. The ITMS intelligently models the skills and knowledge mastered by the student and makes intelligent proactive decisions and notifications based on that model. It also provides intelligent courseware tracking, evaluation, and configuration control. After determining training requirements, it intelligently schedules the required resources. The ITMS is a general tool which can be easily customized to specific domains by end users. Several sites will use the ITMS operationally in Phase 2 to provide feedback and a basis for follow on commercialization.

DESCRIPTORS: *AIR FORCE TRAINING, *ARTIFICIAL INTELLIGENCE, *COMPUTER AIDED INSTRUCTION, MANAGEMENT INFORMATION SYSTEMS, COMBAT READINESS, PERFORMANCE(HUMAN), PROTOTYPES, TRAINING MANAGEMENT, SBIR(SMALL BUSINESS INNOVATION RESEARCH), ITMS(INTELLIGENT TRAINING MANAGEMENT SYSTEM).

♦Included in the DTIC Review, March 2001

AD-A373062

CYBERNET SYSTEMS CORP ANN ARBOR MI

Human Performance-Based Measurement System

28 Dec 1999 201 PAGES

PERSONAL AUTHORS: Braun, Jeff; Lichtenstein, Eric; Lomg, Joe; Le Prell, Glenn; Henry, Martha

ABSTRACT: Fatigue, stress, cognitive overload, and other factors cause errors as human operators perform task operations. The goal of this project was to develop, implement, and test a comprehensive system for measuring and analyzing human performance related data. Applications include basic psycho physiological research, evaluation of computer interfaces, evaluation of other task processes, and real time performance monitoring. The developed system combines measurement and analysis of psychological, physiological, and performance measures into a single system. The combination and correlation of these three factors provides for a more robust and accurate assessment of total human performance. The development effort built off a strong foundation of prior research and development efforts for networked data collection, physiological monitoring, eye tracking, operator workload modeling, and advanced human computer interfaces. The developed software system allows for synchronized collection of data from any number of networked devices and provides an array of signal analysis and display tools to support total performance assessment. The Phase 2 effort developed and implemented the human performance based measurement system and validated operation through a series of performance assessment experiments. The resulting system offers a comprehensive tool set for a wide range of human performance studies.

DESCRIPTORS: *PERFORMANCE(HUMAN), *HUMAN FACTORS ENGINEERING, *PSYCHOPHYSIOLOGY, REAL TIME, COGNITION, DATA ACQUISITION, MAN COMPUTER INTERFACE, WORKLOAD, VIRTUAL REALITY, SBIR(SMALL BUSINESS INNOVATION RESEARCH).

AD-A372438

TRINITY COLL DUBLIN (IRELAND)

1ST International Workshop on Managing Interactions in Smart Environments (MANSE 99)

01 Dec 1999 259 PAGES

PERSONAL AUTHORS: Nixon, Paddy; Lacey, Gerard; Dobson, Simon

ABSTRACT: The Final Proceedings for 1st International Workshop on Managing Interactions in Smart Environments (MANSE 99), 13 December 1999 - 14 December 1999. This is an interdisciplinary conference. The conference will address the convergence of research in Distributed Systems, Robotics and Human Centered Computing within the domain of smart buildings and present a unique opportunity to investigate work that crosses the boundaries of these disciplines.

DESCRIPTORS: *ROBOTICS, *HUMAN FACTORS ENGINEERING, *KNOWLEDGE BASED SYSTEMS, *VIRTUAL REALITY, INTEGRATED SYSTEMS, DISTRIBUTED DATAPROCESSING, PARALLEL PROCESSING, AUTONOMOUS NAVIGATION, INTERACTIVE GRAPHICS, IRELAND, WORKSHOPS, AQ F00-04-0960, PROCEEDINGS.

AD-A371580

DAYTON UNIV OH RESEARCH INST

A Comparison of Virtual and Live Human Standing Reach

01 Oct 1998 26 PAGES

PERSONAL AUTHORS: Nemeth, Kristie J.; Ianni, John D.; Wampler, Jeffrey L.

ABSTRACT: This project investigates the ability of virtual human models to simulate human task performance. A variety of reaching tasks were performed by human subjects and their corresponding virtual human using Transom Jack Software. Transom Jack was able to accurately simulate grasping behaviors for approximately 75% of the trials. The most accurate levels were found at waist and acromion (shoulder) heights. There were significant under estimations for reaches at stature (head) height and significant under estimations for reaches at knee height. Conversely, an overestimation of reach can have more serious implications. In nearly half of the trials at knee height, Transom Jack's simulation outreached the human subjects. Nonetheless, virtual humans provide valuable information in many situations and the technology is rapidly improving.

DESCRIPTORS: *COMPUTER AIDED DESIGN, COMPUTER PROGRAMS, COMPUTERIZED SIMULATION, MODELS, PERFORMANCE(HUMAN), HUMANS, MOTION, ACCURACY, HUMAN FACTORS ENGINEERING, ERGONOMICS, VIRTUAL HUMAN MODELS, TRANSOM JACK SOFTWARE, HFM(HUMAN FIGURE MODELS).

AD-A371142

OHIO STATE UNIV COLUMBUS

Can We Ever Escape from Data Overload? A Cognitive Systems Diagnosis

01 Mar 1998 53 PAGES

PERSONAL AUTHORS: Woods, David D.; Patterson, Emily S.; Roth, Emilie M.

ABSTRACT: Data overload is a generic and tremendously difficult problem. In this report, we diagnose why this is the case and how intelligence analysis presents a particularly difficult version of data overload. We examine three different characterizations that have been offered to capture the nature of the data overload problem and how they lead to different proposed solutions. The first characterization is the clutter problem where there is too much stuff, which leads to proposals to reduce the number of data bits that are displayed. The second characterization is a workload bottleneck where there is too much data to analyze in the time available. Data overload as a workload bottleneck shifts the view to particular activities rather than elemental data and leads to proposals to use automation to perform activities for the practitioner or cooperating automation to assist the practitioner. The third characterization is a problem in finding the significance of data when it is not known a priori what data will be informative. This characterization leads to model based abstractions and representation design techniques as potential solutions By focusing attention on the root issues that make data overload a difficult problem, we have identified a set of challenges that all potential solutions must meet. Most notably, all techniques must deal with the importance of context sensitivity in interpreting data. In order to place data in context, designers need to display data in a conceptual space that depicts the relationships, events, and contrasts that are informative in a field of practice.

DESCRIPTORS: *DATA MANAGEMENT, *COGNITION, *HUMAN FACTORS ENGINEERING, INFORMATION RETRIEVAL, ARTIFICIAL INTELLIGENCE, SYSTEMS ANALYSIS, DATA DISPLAYS.

AD-A370856

NAVAL POSTGRADUATE SCHOOL
MONTEREY CA

The Transfer of Spatial Knowledge from Virtual to Natural Environments as a Factor of Map Representation and Exposure Duration

01 Sep 1999 260 PAGES

PERSONAL AUTHORS: Jones, Quay B.

ABSTRACT: Terrain navigation is a critical skill in the military. Virtual Environments (VEs) have been suggested as a possible tool in training spatial knowledge. However, little research has been conducted into the ability of VEs to impart spatial knowledge of a real world area. This thesis research addresses the utility of VEs to impart spatial knowledge of a natural terrain area compared to traditional methods. Twenty subjects were divided into four training conditions in two experiments. The first experiment had a VE and map only group and trained to a set standard rather than to a time. The second experiment also had a map only and VE group, but trained one hour with a low fidelity map (1:24,000 scale as compared to 1:5,000 scale in earlier experiments). Measures were taken of landmark, route, and survey knowledge. The results suggest that, (1) subjects who trained to standard using a VE demonstrated superior route and landmark knowledge to any other group, (2) spatial ability plays a significant role in navigation performance, and (3) adjusting the fidelity of the map causes individuals to adjust their planned routes to the information that is provided. Furthermore, while good map reading does not guarantee success, poor map reading skills invite failure. Finally, if time is limited, a detailed map is referable to other methods.

DESCRIPTORS: *MAP READING, *VIRTUAL REALITY, *GEOGRAPHICAL INFORMATION SYSTEMS, PERFORMANCE(HUMAN), THESES, TERRAIN INTELLIGENCE, TERRAIN MODELS, TERRAIN FOLLOWING, TRANSFER OF TRAINING.

AD-A370817

NAVAL POSTGRADUATE SCHOOL
MONTEREY CA

The Effect of Presence on the Ability to Acquire Spatial Knowledge in Virtual Environments

01 Sep 1999 125 PAGES

PERSONAL AUTHORS: Bernatovich, David

ABSTRACT: It is unclear what impact presence has on a Virtual Environment's (VE) ability to enhance learning and performance. Currently, there are many theories and conjectures about the effects of presence in VEs. To better the effectiveness of VEs, it is imperative that we determine the impact, both positive and negative, of presence on our ability to perform in VEs. Therefore, we must study how presence affects a person' ability to acquire skills and knowledge. This must include our ability to navigate and perform spatial tasks as well as any other aspect of the real world that may be represented by a VE. To begin understanding how presence affects performance, forty individuals participated in an experiment to determine how presence affects the ability to acquire spatial knowledge in a VE. The purpose of the experiment was to determine if the level of presence in a VE increased or decreased a person's ability to acquire spatial knowledge, to include landmark recognition, procedural knowledge, and survey knowledge. Each participant received one of the following VE treatments: (1) No Sound, (2) Verbal cues with topical information, (3) Verbal cues with spatial information, or (4) a Combination of both topical and spatial information. They were then administered a series of spatial tests. Finally, they were given a presence questionnaire to measure their self-assessed level of presence. The results indicate that as the level of presence in the VE varies, there is no effect on a person's ability to acquire spatial knowledge. A person's spatial performance is more likely the result of their innate spatial abilities and visual memory.

DESCRIPTORS: *PERFORMANCE(HUMAN), *MEMORY(PSYCHOLOGY), *VIRTUAL REALITY, SPATIAL DISTRIBUTION, THESES, VISUAL PERCEPTION, PHYSIOLOGICAL DISORIENTATION, SPACE PERCEPTION, CONDITIONING(LEARNING).

AD-A370517

BDM INTERNATIONAL INC SIERRA VISTA AZ

Modeling Intelligence Production Performance

01 Sep 1999 132 PAGES

PERSONAL AUTHORS: McLean, Marsha B.; Knapp, Beverly G.

ABSTRACT: The objective of this effort was to develop an analysis framework and computer-based tool for simulating and evaluating the impacts of materiel, organizational, and personnel changes in the Military Intelligence (MI) production system. This tool was designed to assist the MI community in assessing new concepts for meeting commander's intelligence requirements of the future. A series of representational models was built first: conceptual, performance, and information quality. The Conceptual Model represented intelligence production as a simple input-process-output model, with nodes representing the functions required to produce intelligence and links representing the information flow. The Performance Model specified the behavioral tasks required to produce intelligence, taxonomy of human performance errors associated with the tasks, and the operational, scenario, and environmental variables that affect task performance. Finally, the Intelligence Quality Model quantified the results of information flow activity and linked the impact of task performance variables when operating on the information. A team of experts in behavioral science, modeling and simulation, and military intelligence built the Intelligence Production Model (IPM). The computer-based IPM was then built by linking these models using a rule-based logic structure and was accessed by a user interface designed to allow analysts to conduct case studies for a wide range of evaluation questions.

DESCRIPTORS: *INFORMATION PROCESSING, COMPUTER PROGRAMS, COMPUTERIZED SIMULATION, MILITARY INTELLIGENCE, INTELLIGENCE, MODELS, PERFORMANCE(HUMAN), INTERFACES, RULE BASED SYSTEMS, INFORMATION THEORY, IPM(INTELLIGENCE PRODUCTION MODEL), COMPUTER BASED LOGIC.

AD-A370316

WASHINGTON UNIV SEATTLE DEPT OF ELECTRICAL ENGINEERING

Stable Haptic Interaction With Virtual Environments

19 Oct 1999 132 PAGES

PERSONAL AUTHORS: Adams, Richard J.

ABSTRACT: A haptic interface is a kinesthetic link between a human operator and a virtual environment. It allows the user of a virtual reality system to feel objects in a virtual world. This dissertation addresses fundamental stability and performance issues associated with haptic interaction. It generalizes and extends the concept of a virtual coupling network, an artificial link between the haptic display and a virtual environment, to include both impedance and admittance models of haptic interaction. A benchmark example is used to expose an important duality between these two cases. Linear circuit theory is employed to develop necessary and sufficient conditions for the stability of a haptic simulation, assuming the human operator and virtual environment are passive. This approach leads to design procedures for virtual coupling networks which give maximum performance while guaranteeing stability.

DESCRIPTORS: *HUMAN FACTORS ENGINEERING, *VIRTUAL REALITY, *TOUCH, PERFORMANCE(HUMAN), THESES, ARTIFICIAL INTELLIGENCE, PERCEPTION(PSYCHOLOGY), STEREOSCOPIC DISPLAY SYSTEMS.

AD-A368698

ARMY RESEARCH INST FOR THE BEHAVIORAL AND SOCIAL SCIENCES ALEXANDRIA VA

High Payoff Tasks for Training Soldiers and Small Unit Leaders in Virtual Environments

01 Sep 1999 21 PAGES

PERSONAL AUTHORS: Pleban, Robert J.; Graham, Scott E.

ABSTRACT: This report describes a multi-tiered process for identifying potential high payoff tasks for training small unit dismounted infantry soldiers in simulated urban operations. Two recently created lists of Infantry tasks and battle drills were evaluated. Four selection criteria were applied: (1) the capability of current and near term individual combatant simulator systems to support specific task related behaviors; (2) the potential transfer effectiveness of practicing these tasks in a virtual environment; (3) the frequency with which task components (behaviors) are performed and; (4) the cost effectiveness/feasibility of performing the task in the virtual environment. Five tasks and five subtasks were retained for subsequent development into training scenarios. The tasks included Assault, Move Tactically, Enter Building and Clear a Room, Reconnoiter Area, and React to Contact. The subtasks included Engage Targets with an M16A1 or M16A2 Rifle, Move as a Member of a Fire Team, Control Movement of a Fire Team, Perform Movement Techniques During MOUT, and Report Information of Potential Intelligence Value. The training scenarios will be evaluated in the Land Warrior Test Bed. These evaluations will help confirm the value of virtual environment simulations as a rehearsal tool for soldiers and small unit leaders.

DESCRIPTORS: *VIRTUAL REALITY, *URBAN WARFARE, *COMBAT SIMULATION, ARMY TRAINING, PERFORMANCE(HUMAN), INFANTRY PERSONNEL, COMPUTER AIDED INSTRUCTION, TRANSFER OF TRAINING, BATTLE MANAGEMENT.

AD-A367691

ADVISORY GROUP FOR AEROSPACE RESEARCH AND DEVELOPMENT NEUILLY-SUR-SEINE (FRANCE)

A Designer's Guide to Human Performance Modelling

01 Dec 1998 162 PAGES

ABSTRACT: Working Group 22 was convened in 1995, jointly sponsored by the Aerospace Medical Panel and the Flight Vehicle Panel to investigate the use of Human Performance Models within the specification, procurement, design, qualification and certification of military systems. In particular the group focused on the selection, application and use of HPMs by the system designer. An expert system approach was selected to ensure that the designer considered all the relevant factors when selecting a new model or tool. This was implemented using a commercially available expert system shell. The user is asked to select options that most closely describe his resources and requirements and the Human Operator Modelling Expert Review (HOMER) then rank orders the HPMs in its database and suggests the most appropriate model. The group carried out some walkthroughs of existing models/tools to demonstrate typical uses in the analysis of specific issues. These are included as case studies. These were included to give potential users some insight into the ease or complexity of use in order to evaluate the required aspect of human performance. In addition the group also considered the model developer community by examining the limitations of existing models, commercial implications and usability issues in order to guide any future development.

DESCRIPTORS: *HUMAN FACTORS ENGINEERING, *EXPERT SYSTEMS, *MAN MACHINE SYSTEMS, NATO, PERFORMANCE(HUMAN), CASE STUDIES, FRANCE, VISUAL PERCEPTION, MAN COMPUTER INTERFACE, SITUATIONAL AWARENESS, FOREIGN REPORTS.

AD-A367429

NATO RESEARCH AND TECHNOLOGY ORGANIZATION NEUILLY-SUR-SEINE (FRANCE)

Alternative Control Technologies (Technologies de Controle non Conventionnelles)

01 Dec 1998 147 PAGES

PERSONAL AUTHORS: Hudgins, Bernard; Leger, Alain; Dauchy, Pierre; Pastor, Dominique; Pongratz, Hans

ABSTRACT: In January 1996, the Working Group 25 of the former AGARD Aerospace Medical Panel began to evaluate the potential of alternative (new and emerging) control technologies for future aerospace systems. The present report summarizes the findings of this group. Through different chapters, the various human factors issues related to the introduction of alternative control technologies into military cockpits are reviewed, in conjunction with more technical considerations of the state of the art of the enabling technologies. Cockpit integration issues are emphasized in regard to both human factors and engineering constraints. Several specific applications of these new technologies in the aerospace environment are considered. Challenges for further developments are identified and recommendations issued. Globally, based upon operational considerations and currently recognized limitations of the HOTAS concept, the conclusion is that Alternative Control Technology should now be progressively introduced into the cockpit, as a function of degree of maturity of the various techniques.

DESCRIPTORS: *COCKPITS, *HUMAN FACTORS ENGINEERING, *ARTIFICIAL INTELLIGENCE, COMPUTERIZED SIMULATION, NATO, FIGHTER AIRCRAFT, INTEGRATED SYSTEMS, NEURAL NETS, ADAPTIVE CONTROL SYSTEMS, STATE OF THE ART, FRANCE, SPEECH RECOGNITION, EYE, MAN MACHINE SYSTEMS, MAN COMPUTER INTERFACE, HEAD(ANATOMY), REMOTELY PILOTED VEHICLES, AUTOMATIC PILOTS, VOICE COMMUNICATIONS, EYE MOVEMENTS, FOREIGN REPORTS, NATO FURNISHED.

AD-A367309

BOSTON UNIV MA CENTER FOR ADAPTIVE SYSTEMS

Self-Organizing Neural Circuits for Sensory-Guided Motor Control

26 Aug 1999 20 PAGES

PERSONAL AUTHORS: Grossberg, Stephen; Bullock, Daniel

ABSTRACT: The reported projects developed mathematical models to explain how self-organizing neural circuits that operate under continuous or intermittent sensory guidance achieve flexible and accurate control of human movement. Neural models were developed for the control of visually guided arm/hand movements, saccadic eye movements, and limb gait transitions. These circuits generate movement trajectories, adapt movement execution on the fly to unforseen contingencies, and improve accuracy over time by learning to act in anticipation of predictable contingencies. The circuits meet behavioral, neuro-biological, and design constraints. Thus, the proposed circuits have operating characteristics that match those documented for human performance and learning, such as voluntary control of speed and amplitude, transfer of learning, and learned recovery from damage to parts of a circuit. The circuits also exhibit stability, robustness, short-term flexibility, and long-term adaptability. The circuits also provide an integrative explanation of many neuro-anatomical, neuro-physiological, and biophysical observations.

DESCRIPTORS: *MATHEMATICAL MODELS, *PERFORMANCE(HUMAN), *HUMANS, *SENSES(PHYSIOLOGY), *SELF ORGANIZING SYSTEMS, VELOCITY, CONTROL, ROBOTICS, SPECIFICATIONS, ACCURACY, ADAPTIVE SYSTEMS, NERVOUS SYSTEM, VISION, CIRCUITS, ADAPTATION, TRAJECTORIES, BEHAVIORAL SCIENCES, EYE MOVEMENTS, BIOPHYSICS, TRANSFER OF TRAINING.

AD-A366993

BOSTON UNIV MA DEPT OF COMPUTER SCIENCE

Head Tracking via Robust Registration in Texture Map Images

01 Aug 1998 8 PAGES

PERSONAL AUTHORS: LaCascia, Marco; Isidoro, John; Sclaroff, Stan

ABSTRACT: A novel method for 3D head tracking in the presence of large head rotations and facial expression changes is described. Tracking is formulated in terms of color image registration in the texture map of a 3D surface model. Model appearance is recursively updated via image mosaicking in the texture map as the head orientation varies. The resulting dynamic texture map provides a stabilized view of the face that can be used as input to many existing 2D techniques for face recognition, facial expressions analysis, lip reading, and eye tracking. Parameters are estimated via a robust minimization procedure; this provides robustness to occlusions, wrinkles, shadows, and specular highlights. The system was tested on a variety of sequences taken with low quality, uncalibrated video cameras. Experimental results are reported.

DESCRIPTORS: *THREE DIMENSIONAL, *PATTERN RECOGNITION, *MAPS, *COMPUTER VISION, *IMAGE REGISTRATION, QUALITY, COLORS, ORIENTATION(DIRECTION), EYE, MAN COMPUTER INTERFACE, PHOTOGRAPHIC IMAGES, FACE(ANATOMY).

AD-A363070

VIRGINIA POLYTECHNIC INST AND STATE UNIV BLACKSBURG DEPT OF COMPUTER SCIENCE

Enhancing a CAVE with Eye Tracking System for Human-Computer Interaction Research in 3D Visualization

03 May 1999 4 PAGES

PERSONAL AUTHORS: Hix, Deborah

ABSTRACT: The objective of this award was to purchase and install two ISCAN Inc. Eye Tracking Systems and associated equipment to create a unique set-up for research in fully immersive virtual environments (VEs), specifically a CAVE. To our knowledge, the Virginia Techinstallation is the first CAVE in the world to have eye-tracking technology incorporated into it We purchased two ISCAN Inc. Eye Tracking Systems, including calibrator, headband-mounted eye imaging system, dual video monitors, line of sight scene imaging system, magnetic head tracker, and appropriate software. The Eye Tracking Systems collect three-dimensional measurements of a user's eye inside the CAVE. These measurements are performed by calculating the direction of a line called the line-of-sight, which originates from a user's eye and extends into the environment in the direction of the pupil's gaze. Intersections between the line-of-sight and visible objects in the VE are computed, and the results are stored to a log file on disk. This work has the potential to lead to strong analytical assessment methodologies for VEs (see Conclusions above) that can reduce the effort and costs of usability evaluations of VEs.

DESCRIPTORS: *VIRTUAL REALITY, *GRAPHICAL USER INTERFACE, IMAGE PROCESSING, LINE OF SIGHT, THREE DIMENSIONAL, VISIBILITY, VISUAL PERCEPTION, VIDEO SIGNALS, HELMET MOUNTED DISPLAYS, EYE TRACKING, VIRTUAL ENVIRONMENTS.

AD-A362882

NAVAL POSTGRADUATE SCHOOL
MONTEREY CA

The Effects of Texture on Distance Estimatation in Synthetic Environments

01 Mar 1999 76 PAGES

PERSONAL AUTHORS: Rowland, James H., III

ABSTRACT: To determine whether egocentric distance judgments are accurate in a virtual environment with different ground surface textures. Observers were immersed within a virtual environment consisting of a large L-shaped room with a column located down one corridor and a flagpole located down the other. The observer's task was to view the column, then turn 90 degrees to view the other corridor where the flag was positioned. The observer then moved the flag's position (by using the joystick) until the distance between the observer and the flag was the same as the distance between the observer and the column. A within-subject design with column size (2 levels), column distance (4 levels), and surface texture (9 levels) was used. The texture beneath the column and the flag was varied from a high-density texture (grass), to medium-density (brick), to a low-density texture pattern (carpet). A within-subject design with column size (2 levels), column distance (4 levels), and surface texture (9 levels) was used. Subjects' distance estimates were significantly better when the brick texture was used underneath the column, than when the grass or carpet texture was used.

DESCRIPTORS: *VISUAL PERCEPTION, *VIRTUAL REALITY, COMPUTERIZED SIMULATION, PERFORMANCE(HUMAN), THESES, MAN COMPUTER INTERFACE, RANGE(DISTANCE), TEXTURE, SPACE PERCEPTION.

AD-A362580

MASSACHUSETTS INST OF TECH
CAMBRIDGE DEPT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

The Finger Walker: A Method to Navigate Virtual Environments

26 May 1998 137 PAGES

PERSONAL AUTHORS: Fitch, Sanford B.

ABSTRACT: There are many factors which make Virtual Environment (VE) systems particularly useful for training applications. Not only can VE systems be easily reconfigured to simulate different real situations, but they can be used to create situations that could not exist in the real world but nonetheless are exceptionally effective in training. Within the general training area, work in this thesis focuses on training directed towards the acquisition of spatial knowledge. There are many cases in which spatial knowledge cannot be acquired in the actual environment, and the training must be accomplished by other means using a VE. A critical factor contributing to the acquisition of spatial knowledge is the method employed for moving around within the VE. Some methods of movement do not provide the user with any easily sensed measure of the amount of effort or work that would be associated with the movement in the real world. This thesis concentrates on the development of an interface that enables the user to "finger walk" through a VE. This interface makes use of a low friction pad that allows the user to finger walk "in place" and an electric field sensing system that monitors the position of the fingers on the pad. The user interface designed effectively tracks the user's movement along the surface of the pad for input into a VE.

DESCRIPTORS: *VIRTUAL REALITY, *GRAPHICAL USER INTERFACE, COMPUTERIZED SIMULATION, SOFTWARE ENGINEERING, INTEGRATED SYSTEMS, REAL TIME, THESES, COMPUTER VISION , VIRTUAL ENVIRONMENTS.

AD-A361027

MISSION RESEARCH CORP FOUNTAIN VALLEY CA

Virtual Prototyping for Personal Protective Equipment and Workplaces

01 Mar 1999 73 PAGES

PERSONAL AUTHORS: Eisler, R. D.; Beecher, R. M.; Tyra, G.; Vaske, D.; O'Keefe, J. A., IV

ABSTRACT: The Phase 1 effort developed a road map for development of a digital human model suitable for virtual prototyping of protective equipment and as a character in a virtual environment. The human model employs detail body contour data obtained from full body scanners onto which an anthropometrically accurate 3D digital model is created. Landmarks on the body surface are used to map internal anatomy. Mass properties of body segments and joint equations of motion are incorporated in order to describe body dynamics of the digital human. Models of protective equipment performance are extended to describe interaction of a range of military projectiles with multi-layer soft fabric body armor, with and without rigid insets. Models of penetrating wounds from ballistic impact, blunt trauma from non-penetrating projectiles, and kinematic trauma will also be incorporated into the digital human model. Finally, a software architecture is developed which includes at least three design cycles. The first cycle is detailed static design which evaluates reduction of joint moment generating capability, thermal load, stability on various terrains, increase in work associated with movement, pressure points under different equipment loads, and casualty reduction potential to prescribed threats. The second cycle is a dynamic analysis, which includes the digital human accomplishing tasks with the equipment in a virtual environment. Parameters relative to equipment performance and the physical state of the virtual humans can be displayed as a function of time. The final cycle is non-virtual field testing.

DESCRIPTORS: *SOFTWARE ENGINEERING, *PROTECTIVE EQUIPMENT, *WORKPLACE LAYOUT, *VIRTUAL REALITY, *VIRTUAL PROTOTYPING, EQUATIONS OF MOTION, COMPUTER AIDED DESIGN, PERFORMANCE(HUMAN), HUMAN BODY, COMPUTER VISION, DIGITAL SIMULATION, ANTHROPOMETRY, SCENE GENERATION.

AD-A360998

SYTRONICS INC DAYTON OH

Low-Level Cognitive Modeling of Aircrew Function Using the Soar Artificial Intelligence Architecture

01 Apr 1998 110 PAGES

PERSONAL AUTHORS: Darkow, David J.; Marshak, William P.; Woodworth, George S.; McNeese, Michael

ABSTRACT: Pilot Vehicle Interface (PVI) testing usually requires extensive Human-In-The-Loop (HITL) simulation. An alternative to HITL testing is to model human computer interaction with an automated cognitive engineering tool. This study used Soar cognitive modeling to compare the effectiveness of an existing and proposed PVI for air-to-ground Maverick missile missions. The baseline interface used a Forward-Looking Infrared Radar (FLIR) to detect and designate targets. The improved PVI had an enhanced FLIR and added Real-Time Information in the Cockpit (RTIC) with annotated overhead imagery of the target area. The Soar software architecture was chosen to model pilot cognition, although target acquisition was more dependent on the pilot's visual and motor functions than cognition. The Soar model accurately predicated faster target acquisition for the RTIC PVI and faster target acquisition for reduced scene complexity. The Soar model correctly indicated that increased scene complexity caused larger increases in target acquisition time for the RTIC PVI condition as compared to the baseline condition (HITL 179% increase, Soar 47% increase). Furthermore, Soar was the only model that accurately predicted increased latency in the RTIC condition while both Cognitive and Traditional Task Analyses predicted decreased latencies.

DESCRIPTORS: *HUMAN FACTORS ENGINEERING, *ARTIFICIAL INTELLIGENCE, *MAN COMPUTER INTERFACE, COMPUTERIZED SIMULATION, SOFTWARE ENGINEERING, OPTICAL RADAR, FORWARD LOOKING INFRARED SYSTEMS, COCKPITS, REAL TIME, PERFORMANCE(HUMAN), TARGET ACQUISITION, COGNITION, FLIGHT SIMULATION, AIR TO SURFACE MISSILES, SOAR COMPUTER PROGRAM, PVI(PILOT VEHICLE INTERFACE).

AD-A360838

NATO RESEARCH AND TECHNOLOGY ORGANIZATION NEUILLY-SUR-SEINE (FRANCE)

Sensor Data Fusion and Integration of the Human Element (la Fusion de donnees de senseur et l'integration du facteur humain)

01 Feb 1999 236 PAGES

ABSTRACT: Partial contents include: (1) On the Design of a Decision Support System for Data Fusion and Resource Management in a Modern Frigate; (2) Measures of Merit for Collaborative Collection, Connection, and Execution Management; (3) Contextual Information and Multi-sensor Data Fusion for Battlefield Applications Image Data Fusion for Future Enhanced Vision Systems; (4) A Distributed System for Command and Control Applications with Programming Language Abstraction; (5) Novel Concepts for Identity Fusion in an Air Combat Environment; (6) Integration of the Human Operator into Complex Air Systems Using Alternative Control Technologies; (7) Fusion and Display of Data According to the Design Philosophy of Intuitive Use; (8) Enhanced and Synthetic Vision System Concept for Application to Search and Rescue Missions Fusion and Display of Tactical Information Within Battlefield Helicopters; (9) The Multi-Sensor Integration System for NATO E-3A Mid-Term Modernization; (10) Environment Perception Process in Maritime Command and Control Utilizing CORBA Concepts for Command and Control Systems; and (11) Integrating Voice Recognition and Automatic Target Cueing to Improve Aircrew-System Collaboration for Air-to-Ground Attack.

DESCRIPTORS: *COMMAND AND CONTROL SYSTEMS, *DATA FUSION, *MAN MACHINE SYSTEMS, *SENSOR FUSION, IMAGE PROCESSING, INTEGRATED SYSTEMS, SYMPOSIA, TARGET RECOGNITION, FRANCE, SPEECH RECOGNITION, DATA DISPLAYS, MULTISENSORS, TACTICAL DATA SYSTEMS, DECISION SUPPORT SYSTEMS, AQ F99-06-1172, FOREIGN REPORTS.

AD-A358600

AIR FORCE INST OF TECH WRIGHT-PATTERSONAFB OH SCHOOL OF ENGINEERING

Feature Saliency in Artificial Neural Networks with Application to Modeling Workload

01 Dec 1998 335 PAGES

PERSONAL AUTHORS: Greene, Kelly A.

ABSTRACT: This dissertation research extends the current knowledge of feature saliency in Artificial Neural Networks (ANN). Feature saliency measures allow for the user to rank order the features based upon the saliency, or relative importance, of the features. Selecting a parsimonious set of salient input features is crucial to the success of any ANN model. In this research, several methodologies were developed using the Signal to Noise Ratio (SNR) Feature Screening Method and its associated SNR Saliency Measure for selecting a parsimonious set of salient features to classify pilot workload in addition to air traffic controller workload. Candidate features were derived from electroencephalography (EEG), electrocardiography (EKG), electro-oculography (EOG), and respiratory gauges. In addition, a new saliency measure was developed that can account for time in Elman Recurrent Neural Networks (RNN). This Partial Derivative Based Spatial Temporal Saliency Measure is used via a Spatial Temporal Feature Screening Method for selecting a parsimonious set of salient features in both time and space. Finally, a technique for investigating the memory capacity of an Elman RNN was developed.

DESCRIPTORS: *NEURAL NETS, *ARTIFICIAL INTELLIGENCE, *MENTAL ABILITY, PERFORMANCE(HUMAN), PILOTS, SIGNAL TO NOISE RATIO, THESES, WORKLOAD, AIR TRAFFIC CONTROLLERS, ELECTROENCEPHALOGRAPHY, ELECTROCARDIOGRAPHY.

AD-A358413

MICHIGAN UNIV ANN ARBOR KRESGE HEARINGRESEARCH INST

Virtual Auditory Space: Individual Differences

01 Mar 1998 4 PAGES

PERSONAL AUTHORS: Middlebrooks, J. C.

ABSTRACT: Head Related Transfer Functions (HRTFs) capture the direction dependent filter characteristics of the external ears. When a sound is filtered by HRTFs measured from a listener's own ears and played over headphones, the listener hears a virtual source that is well localized in space. When sounds were filtered by other listeners' HRTFs, listeners showed fairly accurate localization in the lateral dimension but showed conspicuous vertical and front/back errors. We examined differences among HRTFs measured from 45 listeners. We quantified differences by subtracting HRTFs between listeners for corresponding locations, then computing the variance of the resulting difference spectra across 393 locations. Inter listener differences could be reduced by shifting HRTFs scaling in frequency. Optimal scalars reduced variances by an average of 20.2% across all pairs of listeners and by more than 50% in 9.5% of listener pairs. The optimal scalar for any pair of listeners correlated highly with the relative sizes of certain physical dimensions. When HRTFs were shifted optimally then used in virtual localization trials, all measures of virtual localization performance tended to improve. In the majority of cases, the performance penalty for use of HRTFs from another listener was reduced.

DESCRIPTORS: *VIRTUAL REALITY, *AUDITORY PERCEPTION, PERFORMANCE(HUMAN), HUMAN FACTORS ENGINEERING, ACOUSTIC SIGNALS, AUDITORY ACUITY, EARPHONES, HRTF(HEAD RELATED TRANSFER FUNCTIONS).

AD-A357986

TRANSOM TECHNOLOGIES INC ANN ARBOR MI

Human Dynamics Modeling Topic 97.2, A97-O24

06 Jun 1998 12 PAGES

PERSONAL AUTHORS: Schutte, Lisa; Young, Michael

ABSTRACT: The purpose of the Phase 1 research was to investigate and prototype methods to simulate realistic human activities utilizing physics based motions and limb trajectory control strategies. Under the Phase 1 work, researchers successfully simulated dynamic motion, impacts, and control schemes utilizing an advanced human model and 3D computer graphics program (Transom Jack). A comprehensive survey of potential control schemes was completed and documented, and the most promising control method was further developed. Potential applications for this work include dynamic simulation of human movement and locomotion under varying timing and loading conditions. The combination of 3D graphics, a realistic human biomechanical model, physics based motion, and dynamic control capabilities will allow assessment of energy expenditure, joint loading, and man equipment interface issues.

DESCRIPTORS: *COMPUTERIZED SIMULATION, *MAN MACHINE SYSTEMS, *BIOMECHANICS, PERFORMANCE(HUMAN), PROTOTYPES, HUMAN FACTORS ENGINEERING, MAN COMPUTER INTERFACE, LOCOMOTION, SBIR(SMALL BUSINESS INNOVATION RESEARCH).

AD-A357190

WRIGHT STATE UNIV DAYTON OH DEPT OF PSYCHOLOGY

Instrumentation to Enhance DoD-Relevant Research on Cognitive Workload in UAVs, Image, Exploitation, and Spatial Hearing

01 Feb 1998 10 PAGES

PERSONAL AUTHORS: Gilkey, Robert H.

ABSTRACT: The project goals have been to provide enhanced real-time graphics generation capacity, computational power, and real-time audio signal processing capability for the Virtual Environment Research, Interactive Technology, And Simulation (VERITAS) facility, making it better suited to the demands of DoD-relevant research projects on human performance in complex environments. VERITAS is owned by Wright State University, but housed at Wright-Patterson AFB. It includes a CAVE(Trademark), which is an immersive, wide field-of-view, stereoscopic, real-time interactive display system, allowing the user to move through virtual environments with minimal encumbrances. The CAVE(Trademark) is controlled by a Silicon Graphics Onyx(trademark) computer with Infinite Reality(Trademark) graphics. The high-fidelity simulations in this facility allow a variety of questions related to human effectiveness to be addressed. The DURIP funds were used to purchase a multiprocessor computational subsystem, a graphics generation subsystem, and an acoustics generation subsystem. These subsystems provide critical capabilities for computationally intensive, real-time-constrained applications, including simulation, virtual environments, auditory and visual displays, motor control, and human perception and cognition. This instrumentation has supported specific funded DoD projects investigating: (1) display and control representations for UAV operation, and (2) binaural and spatial hearing.

DESCRIPTORS: *COMPUTERIZED SIMULATION, *DISPLAY SYSTEMS, *VIRTUAL REALITY, *VERITAS(VIRTUAL ENVIRONMENT RESEARCH INTERACTIVE TECHNOLOGY AND SIMULATION), REAL TIME, TEST EQUIPMENT, INTERACTIVE GRAPHICS, STEREOSCOPIC DISPLAY SYSTEMS, HAPTIC DISPLAYS, UAV(UNMANNED AERIAL VEHICLES).

AD-A356098

NAVAL POSTGRADUATE SCHOOL MONTEREY CA

Two-Handed, Whole-Hand Interaction

01 Sep 1998 96 PAGES

PERSONAL AUTHORS: Cockayne, William R.

ABSTRACT: This thesis investigates the application of Human Ability Requirements (HARs) to problem of two handed, whole handed interaction. The methodology is derived from the use of HARs in the world of human performance evaluation. This research is based onthe need to understand how humans perform tasks in order to guide the understanding of the requirements of advanced interface technology development. The thesis presents the background for these two areas of research, taxonomies and whole hand interaction. It goes on to develop a taxonomy and classification of two handed, whole hand interaction for the real world and virtual environments. This taxonomy is used to analyze a large number of real world tasks, to further the development of a series of tests to externally validate the classification, and to analyze the tasks of the 91B Field Medic. This thesis further presents recommendations for how this methodology can be used to develop taxonomies for other areas of human interaction, for how this taxonomy can be used by researchers and practitioners, and areas of further research related to both areas.

DESCRIPTORS: *HUMAN FACTORS ENGINEERING, *VIRTUAL REALITY, *GRAPHICAL USER INTERFACE, COMPUTERIZED SIMULATION, SKILLS, PERFORMANCE(HUMAN), THESES, TOUCH.

AD-A355911

NATO RESEARCH AND TECHNOLOGY ORGANIZATION NEUILLY-SUR-SEINE (FRANCE)

Alternative Control Technologies: Human Factors Issues

01 Oct 1998 114 PAGES

ABSTRACT: With the increasing intelligence of computer systems, it is becoming more desirable to have an operator communicate with machines rather than simply operate them. In combat aircraft, this need to communicate is made quite crucial due to high temporal pressure and workload during critical phases of the flight (ingress, engagement, deployment of self-defense). The HOTAS concept, with manual controls fitted on the stick and throttle, has been widely used in modern fighters such as F16, F18, EFA and Rafale. This concept allows pilots to input real time commands to the aircraft system. However, it increases the complexity of the pilot task due to inflation of real time controls, with some controls being multifunction. It is therefore desirable, in the framework of "ecological interfaces", to introduce alternative input channels in order to reduce the complexity of manual control in the HOTAS concept and allow more direct and natural access to the aircraft systems. Control and display technologies are the critical enablers for these advanced interfaces. Careful design and integration of candidate control technologies will result in human-machine interfaces which are natural, easier to learn, easier to use, and less prone to error. Significant progress is being made on using signals from the brain, muscles, voice, lip, head position, eye position and gestures for the control of computers and other devices. Judicious application of alternative control technologies has the potential to increase the bandwidth of operator-system interaction, improve the effectiveness of military systems, and realize cost savings.

DESCRIPTORS: *HUMAN FACTORS ENGINEERING, *JET FIGHTERS, *SPEECH RECOGNITION, *ARTIFICIAL INTELLIGENCE, COMPUTERIZED SIMULATION, INTEGRATED SYSTEMS, ADAPTIVE CONTROL SYSTEMS, COMPUTER AIDED DESIGN, COCKPITS, PILOTS, FRANCE, MAN MACHINE SYSTEMS, MAN COMPUTER INTERFACE, VOICE COMMUNICATIONS, EYE MOVEMENTS.

AD-A355542

NAVAL POSTGRADUATE SCHOOL MONTEREY CA

Spatial Knowledge Acquisition and Transfer from Virtual to Natural Environments for Dismounted Land Navigation

01 Sep 1998 311 PAGES

PERSONAL AUTHORS: Goerger, Simon R.

ABSTRACT: Navigation and terrain familiarity are critical for mission success in the military. Virtual environments (VEs) have often been suggested as a useful tool in addressing these issues. This thesis research addresses the utility of VEs to improve spatial knowledge of and navigation performance through natural terrain compared to traditional methods. In this experiment, fifteen subjects were assigned to one of three training conditions. The map group studied the environment using only an orienteering map. The real world group studied the environment using the map and explored the actual terrain. The VE group studied the terrain using both the map and a real-time VE. Measures were taken of both route and configuration knowledge. The results suggest four conclusions. First, training conditions have no statistically significant effect on an individual's ability to obtain and demonstrate spatial knowledge of a natural environment Second, spatial ability plays a significant role in navigation performance. Third, exposure to the actual terrain or to a virtual representation of the terrain seems to eliminate ambiguities in an individual's mental map by providing dynamic imagery to clarify propositional knowledge gained from maps. However, this factor has not been shown to improve performance by the measures used here.

DESCRIPTORS: *VIRTUAL REALITY, *SURFACE NAVIGATION, *TERRAIN MODELS, COMPUTERIZED SIMULATION, REAL TIME, PERFORMANCE(HUMAN), THESES, HIGH RESOLUTION, DATA ACQUISITION, MAN COMPUTER INTERFACE, MENTAL ABILITY, SPACE PERCEPTION, TERRAIN FOLLOWING, VIRTUAL ENVIRONMENTS.

AD-A355474

NAVAL POSTGRADUATE SCHOOL
MONTEREY CA

Auditory-Visual Cross-Modal Perception Phenomena

01 Sep 1998 275 PAGES

PERSONAL AUTHORS: Storms, Russell L.

ABSTRACT: The quality of realism in virtual environments is typically considered to be a function of visual and audio fidelity mutually exclusive of each other. However, the virtual environment participant, being human, is multi-modal by nature. Therefore, in order to more accurately validate the levels of auditory and visual fidelity required in a virtual environment, a better understanding is needed of the inter-sensory or cross modal effects between the auditory and visual sense modalities. To identify whether any pertinent auditory visual cross modal perception phenomena exist, 108 subjects participated in three main experiments which were completely automated using HTML, Java, and JavaScript computer programming languages. Visual and auditory display quality perception were measured intramodally and intermodally by manipulating visual display pixel resolution and Gaussian white noise level and by manipulating auditory display sampling frequency and Gaussian white noise level. Statistically significant results indicate that: (1) medium or high quality auditory displays coupled with high quality visual displays increase the quality perception of the visual displays relative to the evaluation of the visual display alone, and (2) low quality auditory displays coupled with high quality visual displays decrease the quality perception of the auditory displays relative to the evaluation of the auditory display alone. These findings strongly suggest that the quality of realism in virtual environments must be a function of both auditory and visual display fidelities inclusive of each other.

DESCRIPTORS: *VISUAL PERCEPTION, *VIRTUAL REALITY, *AUDITORY PERCEPTION, WHITE NOISE, GAUSSIAN NOISE, THESES, HUMAN FACTORS ENGINEERING, MAN COMPUTER INTERFACE.

AD-A353750

TEXAS UNIV AT AUSTIN CENTER FOR VISION AND IMAGE SCIENCES

Local Spatio-Temporal Analysis in Vision Systems

01 Sep 1998 29 PAGES

PERSONAL AUTHORS: Geisler, W. S.; Bovik, A. C.; Cormack, L. K.; Gilden, D. L.; Super, B. J.

ABSTRACT: This is the final progress report of the vision group at the University of Texas under support of AFOSR URI grant F49620-93-1-0307. In this report we will attempt to summarize the major accomplishments over the previous 5 years. Aim 1: to develop a mathematical model of the initial stages of visual processing (the front end mechanisms), based upon a wide range of physiological and psychophysical data. Aim 2: To develop new methods and models of local frequency coding. Aim 3: To develop new mathematical models and computer-vision algorithms for performing complex visual tasks that are based upon local frequency coding representations. Aim 4: To develop models for human performance in complex visual tasks that build upon current understanding of the front-end mechanisms. Aim 5: To develop a computational test bed for implementing, comparing, integrating and visualizing the different models and modules developed during the project, using a massively parallel machine and graphics workstation front-end.

DESCRIPTORS: *MATHEMATICAL MODELS, *COMPUTER VISION, *VISUAL PERCEPTION, ALGORITHMS, FREQUENCY, IMAGE PROCESSING, TEST BEDS, COMPUTATIONS, PERFORMANCE(HUMAN), CODING, VISION, RANGE(EXTREMES), TEXAS, PSYCHOPHYSICS.

AD-A353096

ANACAPA SCIENCES INC SANTA BARBARA CA

Combat Mission Training Research at the 58th Special Operations Wing: A Summary

01 Jul 1998 59 PAGES

PERSONAL AUTHORS: Spiker, V. A.; Nullmeyer, Robert T.; Tourville, Steven J.; Silverman, Denise R.

ABSTRACT: This report summarizes three empirical studies conducted at Kirtland Air Force Base during 1995-1997. The first study examined the relationship between Crew Resource Management (CRM) processes and mission performance for MC-130P Combat Shadow crews who were receiving annual simulator refresher training. Using independent assessments of process and performance, a strong, positive correlation (r=.86) was observed between CRM effectiveness at the crew-level and their performance during a simulated tactical mission. A strong association between the quality of a crew's mission planning activities and subsequent mission performance (r=.60)was also observed. A second study investigated human factors characteristics of an aerial gunner/scanner simulator (AGSS) recently installed at the 58th Special Operations Wing. The AGSS is a virtual reality (VR) training device that uses a CRT-based, helmet-mounted display and a three degree-of-freedom motion base to train rotary-wing gunners and scanners. A usability assessment by 11 aerial gunner instructors showed that while the devices' VR properties have enormous training potential, the device's human factors aspects need improvement, including the CRTs, head tracker, fitting procedures, and cables. A third study explored the impact of networked simulation on combat mission training. Ninety-nine crewmembers participating in nine networked training exercises were surveyed following training in which MH-53J, MH-60G, TH-53A, and MC-130P weapon system trainers were linked. Survey results strongly support the value of networked training in such areas as multi-ship tactics, aerial refueling operations, formation flight, situation awareness, and mission team coordination. Areas in need of improvement include establishing training objectives, incorporating emergency procedures into the scenario, and leveling the task demands across crew positions and weapon systems.

DESCRIPTORS: *COMPUTERIZED SIMULATION, *FLIGHT CREWS, *TRAINING DEVICES, FLIGHT TRAINING, PERFORMANCE(HUMAN), OPERATIONAL EFFECTIVENESS, RESOURCE MANAGEMENT, MILITARY TRAINING, FLIGHT SIMULATORS, ANNUAL REFRESHER TRAINING, COMBAT TRAINING, CRM(CREW RESOURCE MANAGEMENT).

AD-A352026

ARMY RESEARCH INST FOR THE BEHAVIORAL AND SOCIAL SCIENCES ALEXANDRIA VA

Effect of a Body Model on Performance in a Virtual Environment Search Task

01 Aug 1998 37 PAGES

PERSONAL AUTHORS: Singer, Michael J.; Ehrlich, Jennifer A.; Allen, Robert C.

ABSTRACT: The U.S. Army Research Institute is investigating requirements for using Virtual Environments (VE) in training dismounted soldiers. This experiment investigated full body representation (generic) versus a hand linked pointer on movement performance in an office building interior during a search task. The search task was used as a representative dismounted soldier activity in urban environments. The VE used a biocular Head Mounted Display (HMD) with head coupled and body referenced movement control. Sensors enabled participants to walk through the VE while performing the search task in six repeated trials. Movement time and number of collisions during discrete phases of the search task revealed no significant differences found between full body and pointer representations, although significant improvement was found over repeated trials. Field of view is discussed as a possible intervening aspect. A Simulator Sickness Questionnaire (SSQ) was administered before, during, immediately after the experiment, and after a recovery period. Significant changes in the SSQ were found over the course of the experiment, but were not related to the body representation condition. The results indicate a rapid onset of symptoms followed by some adaptation to the VE, and rapid recovery. The Immersive Tendencies Questionnaire administered pre-experiment, and the Presence Questionnaire administered post-experiment, were not significantly related to the body representation conditions.

DESCRIPTORS: *PERFORMANCE(HUMAN), *VIRTUAL REALITY, *HEAD UP DISPLAYS, ARMY PERSONNEL, ARMY TRAINING, HUMAN FACTORS ENGINEERING, MOTION SICKNESS, SPACE PERCEPTION, ADJUSTMENT(PSYCHOLOGY).

AD-A350481

AIR FORCE RESEARCH LAB WRIGHT-PATTERSON AFB OH HUMAN EFFECTIVENESS DIRECTORATE

Building the LeM2*R3 Model of Pilot Trust and Dynamic Workload Allocation. A Transition of Theory and Empirical Observations to Cockpit Demonstration

01 Feb 1998 102 PAGES

PERSONAL AUTHORS: Raeth, Peter G.; Reising, John M.

ABSTRACT: For pilots to accept active decision aids during complex flight scenarios, it is essential that the automation work is in synergy with aircrew. To accomplish this, the automation must go well beyond menu and macro selections, where the pilot must explicitly tell the automation what to do and when to do it. It must also transcend "mother may I" approaches, where the automation asks for permission to proceed. To these traditional barriers, the automation needs a sense of how the pilot will react in a given situation and, based on that reaction, how much of the workload could be allocated to the automation at any given time. For this purpose, the authors reviewed the literature on human factors and dynamic function allocation. This literature provided a wealth of information on this topic. Based on the current state of the art in this topic area, the authors developed and tested a dynamic model of pilot trust and workload allocation. This "full degrees of freedom" model transitions human factors theory, as it exists today, into an engineering application. The resulting model can be combined with other information obtained from static and continuous processes to divide the workload and minimize cognitive overload.

DESCRIPTORS: *DECISION MAKING, *COCKPITS, *HUMAN FACTORS ENGINEERING, *ARTIFICIAL INTELLIGENCE, COMPUTERIZED SIMULATION, AUTOMATION, COMPUTER AIDED DESIGN, HEURISTIC METHODS, SYSTEMS ANALYSIS, MAN COMPUTER INTERFACE, WORKLOAD, FLIGHT SIMULATION, DECISION AIDS, UCAV(UNINHABITED COMBAT AERIAL VEHICLE).

AD-A349708

BATTELLE DAYTON OPERATIONS OH

A Survey of Immersive Technology For Maintenance Evaluations

01 Apr 1998 74 PAGES
PERSONAL AUTHORS: Stonum, Ron

ABSTRACT: Virtual reality has come a long way since the turn of the decade and is continually growing, with extensive research by commercial, government, and educational agencies. This paper describes current virtual reality technologies available; undergoing virtual reality research & development; and issues related to those technologies that can be applied to simulated maintenance evaluations. There are still hardware and software issues that need to be overcome before virtual presence is truly felt. These issues include simulator sickness where the inner ear cannot feel the translation motion unless it is physically experienced; and multi-person tasks where peripheral vision plays a major role in the accomplishment of the task.

DESCRIPTORS: *MAINTENANCE, *VIRTUAL REALITY, COMPUTERIZED SIMULATION, COMPUTER AIDED DESIGN, HUMAN FACTORS ENGINEERING, MAN MACHINE SYSTEMS, MOTION SICKNESS, HELMET MOUNTED DISPLAYS, PERIPHERAL VISION.

AD-A348364

ARIZONA STATE UNIV TEMPE

The Effects of Display Type and Spatial Ability on Performance During a Virtual Reality Simulation

01 Aug 1998 104 PAGES

PERSONAL AUTHORS: Manrique, Fernando

ABSTRACT: The purpose of this study was to investigate the effects of display type and spatial ability level on performance during three trials of a Virtual Reality (VR) simulation. Seventy-six Air Force Reserve Officer Training Corps cadets were identified as having high or low spatial ability level. Subjects used either a Helmet-Mounted Display (HMD) or a standard computer monitor to perform a VR simulation. The study examined the effects of display type, spatial ability, and number of trials on simulation performance, performance strategy, discomfort level, display characteristics and attitude. Results indicated that subjects with high spatial ability performed significantly better on the simulation than subjects with low spatial ability. Performance results revealed a significant interaction between spatial ability level and trial. Analyses for each spatial ability group showed that the performance of high spatial ability subjects improved significantly from trial to trial. The performance of low spatial ability subjects did not significantly improve. There were no significant performance differences between subjects that wore the HMD versus subjects that did the simulation on a standard computer monitor. Results for performance strategy suggested that high spatial ability subjects used more effective strategies during the simulation than low spatial ability subjects. Exploratory data analysis indicated that performance strategy was a stronger predictor of performance than spatial ability. Discomfort survey results indicated that subjects who used the HMD reported significantly higher levels of cold sweating and difficulty focusing than subjects who used the standard computer monitor. HMD subjects also rated display resolution significantly lower than computer monitor subjects.

DESCRIPTORS: *VIRTUAL REALITY, *HELMET MOUNTED DISPLAYS, *HEAD UP DISPLAYS, *SPACE PERCEPTION, SPATIAL DISTRIBUTION, PERFORMANCE(HUMAN), PSYCHOMOTOR FUNCTION, THESES, HUMAN FACTORS ENGINEERING, PERSPIRATION.

AD-A347740

ARMY RESEARCH LAB ABERDEEN PROVING GROUND MD HUMAN RESEARCH AND ENGINEERINGDIRECTORATE

A Comparison of Various Types of HeAD-Related Transfer Functions for 3-D Sound in the Virtual Environment

01 May 1998 24 PAGES

PERSONAL AUTHORS: Savick, Douglas S.

ABSTRACT: Simulation using virtual reality (VR) is becoming an effective tool for the Army in training soldiers to do their required tasks. In VR, the human operator can interact with a wide variety of computer generated worlds developed from real or imaginary scenarios or both. The training that a soldier receives by simulation is usually cost effective to the Army and in a number of cases is safer for the individual than training in the real environment. Three dimensional (3-D) sound in the Virtual Environment (VE) provides a more realistic simulation of acoustic environments compared to diotic (mono) or dichotic (stereo) sound presentation. The major benefit of using 3-D sound is that an individual can determine the sound source direction. When sounds that are perceived to have direction and sights that represent virtual objects that produce the sounds are provided through a head mounted display, a person can monitor and identify sources of information from all possible locations. The purpose of this study was to determine if 3-D sound generated by a 3-D sound system could enhance the realism or fidelity of the VE. The main objective of the study was to determine if an individual could distinguish the direction of a sound source within a reasonable degree of accuracy. Three dimensional sound is produced by using a mathematical representation of the filtering characteristics of the pinnae provided through Head Related Transfer Functions (HRTFs). The HRTFs can be developed by recording a generated broadband sound using a probe microphone in the ear canal and subsequently dividing the Fourier transform of the recorded sound by that of the generated sound. When digital filtering techniques are used, HRTFs can be applied to sounds through headphones. When an arbitrary sound is filtered with HRTF based filters, the sound should appear to come from specified virtual locations outside the earphones.

DESCRIPTORS: *ARMY TRAINING, *VIRTUAL REALITY, *AUDITORY SIGNALS, *AUDITORY PERCEPTION, COMPUTERIZED SIMULATION, ACOUSTIC FILTERS, DIGITAL FILTERS, HEAD UP DISPLAYS, PSYCHOACOUSTICS, AUDITORY ACUITY, EARPHONES, SOUND GENERATORS, HRTF(HEAD RELATED TRANSFER FUNCTIONS).

AD-A345970

NAVAL POSTGRADUATE SCHOOL
MONTEREY CA

An Assessment of the Shipboard Training Effectiveness of the Integrated Damage Control Training Technology (IDCTT) Version 3.0

01 Mar 1998 166 PAGES

PERSONAL AUTHORS: Coughlin, Stephen J.

ABSTRACT: The ability of a ship's crew to control damage is a critical measure of readiness for U.S. Navy ships. Proficiency in this area is largely a function of routine shipboard training. Since damage control skills tend to be perishable if not continuously practiced, shipboard personnel must have an effective means of exercising damage control skills. Computer based technologies that utilize the advantages of Interactive Courseware (ICW) present training opportunities that challenge the traditional methods of shipboard training. The Integrated Damage Control Training Technology (IDCTT) is an application of ICW that allows shipboard repair teams to exercise their damage control skills continuously. The trainer was installed onboard USS Harpers Ferry (LSD-49) and evaluated as a stand alone training device through administration of opinion surveys and comparison to various aspects of full scale drills with a standardized performance evaluation system. The shipboard IDCTT was found to be an effective shipboard training device that saves time. Additionally, it has significant cross training and team building qualities that can be integrated into an existing damage control training program.

DESCRIPTORS: *COMPUTER AIDED INSTRUCTION, *BOAT AND SHIP SAFETY, *DAMAGE CONTROL, *DISTRIBUTED INTERACTIVE SIMULATION, SKILLS, SURVIVABILITY, TRAINING DEVICES, OPERATIONAL READINESS, PERFORMANCE(HUMAN), THESES, SHIP PERSONNEL, NAVAL TRAINING, IDCTT(NTEGRATED DAMAGE CONTROL TRAINING TECHNOLOGY).

AD-A344917

ARMY RESEARCH INST FOR THE BEHAVIORAL AND SOCIAL SCIENCES FORT BENNING GA

Proposed Army Research Institute Support for Army After Next Experimental Unit

01 May 1998 40 PAGES

PERSONAL AUTHORS: Graham, Scott E.

ABSTRACT: The Army is discussing the creation of an experiment unit that can be used to evaluate and refine concepts being developed for the Army After Next (AAN). The purpose of this scripted briefing is to describe what the Army Research Institute (ARI) could do in support of an AAN Experiment Unit (EX Unit), should such an organization be established. The recommendations are based on well-established military psychology principles derived from decades of behavioral science research. In addition, critical research issues are identified that we believe need to be addressed. ARI is prepared to help lead in the design and the utilization of the EX Unit. Our proposed effort uses a systems approach to organize, understand, and address AAN training and personnel performance issues. Among the components to be proposed are sequential selection, assignment, and training systems or subsystems. These systems will rely heavily on the development and use of virtual and constructive simulation environments for concept development and evaluation. Virtual prototypes of future weapon systems and mixes, Communication patterns, and organizational structures will need to be constructed as a means to empirically determine effective if not optional, job structures, personnel requirements, skill mixes, communication patterns, and Tactics, Techniques and Procedures (TTPs). Much of the focus is on enhancing the collective performance of AAN teams.

DESCRIPTORS: *SIMULATION, *ARMY RESEARCH, *MILITARY TRAINING, *ARMY PLANNING, *AAN(ARMY AFTER NEXT), MILITARY REQUIREMENTS, PERFORMANCE(HUMAN), WEAPON SYSTEMS, SYSTEMS APPROACH, PERSONNEL SELECTION, VIRTUAL REALITY, MILITARY PSYCHOLOGY, BEHAVIORAL SCIENCES, SPECIAL OPERATIONS FORCES, EXPERIMENTAL FORCE.

AD-A344313

NOTTINGHAM UNIV (UNITED KINGDOM)

Proceedings of the Second European Conference on Cognitive Modelling

04 Apr 1998 218 PAGES

PERSONAL AUTHORS: Ritter, Frank E.; Young, Richard M.

ABSTRACT: This interdisciplinary conference covered all areas of cognitive modeling, including artificial intelligence programming; classification; problem solving; reasoning; inference; learning; language processing; human computer interaction; symbolic and connectionist models; evolutionary computation; artificial neural networks; grammatical inferences; reinforcement learning; and data sets designed to test models.

DESCRIPTORS: *NEURAL NETS, *COGNITION, *HUMAN FACTORS ENGINEERING, SYMPOSIA, REASONING, COMPUTER PROGRAMMING, PROBLEM SOLVING, ARTIFICIAL INTELLIGENCE, MAN COMPUTER INTERFACE, BIONICS, CONDITIONING(LEARNING), UNITED KINGDOM, FOREIGN REPORTS, PROCEEDINGS.

♦ **AD-**A342328

COMPUTER GRAPHICS SYSTEMS DEVELOPMENT CORP MOUNTAIN VIEW CA

Force/Tactile Feedback System for Virtual Reality Environments

03 Apr 1998 51 PAGES

ABSTRACT: TOPIT- Touch Objects Positioned In Time, is a virtual reality system that allows a user, wearing a Head Mounted Display (HMD) to actuate control knobs or dials on a cockpit instrument panel appearing in user's HMD. The actual knobs, dials, and control are delivered to the user by an x-y-z robot. The instrument panel can be reconfigured entirely in software. A tracker and data glove continually provide the position of user's hand and finger to a computer, and the computer commands the servomechanism system to place the correct type of control in the correct position to be actuated. The servo system has a 'touch panel' that contains examples of a dozen or so different types of controls, such as toggle switches, knobs, and push buttons that are used repeatedly to represent any number of instrument panel controls. One key aspect of the system is building a servo system that moves fast enough to put the knobs, dials, and control in place at the time when the user reaches for it. Another key aspect is achieving precise low-latency tracking of both user's head and user's hand.

DESCRIPTORS: *ROBOTICS, *VIRTUAL REALITY, *HELMET MOUNTED DISPLAYS, COCKPITS, MAN MACHINE SYSTEMS, MAN COMPUTER INTERFACE, MULTISENSORS, SERVOMECHANISMS, HEAD UP DISPLAYS, INSTRUMENT PANELS, CONTROL KNOBS, CONTROL STICKS, SBIR(SMALL BUSINESS INNOVATION RESEARCH), TOPIT(TOUCH OBJECTS POSITIONED IN TIME).

♦ Included in the DTIC Review, March 2001

AD-A341712

ARMY RESEARCH LAB ABERDEEN PROVING GROUND MD

Skill Level 10 Navigational Skills: An Examination of Tactical Unmanned Vehicle (TUV) Soldier-Marine Capabilities

01 Mar 1998 30 PAGES

PERSONAL AUTHORS: Scribner, David R.

ABSTRACT: An analysis was performed to identify specific skills required to successfully perform mission planning and navigational tasks for the future tactical unmanned vehicle (TUV) and to determine if U.S. Army soldiers and U.S. marines with a beginning skill level of 10 have those skills. This analysis was performed by the Human Research and Engineering Directorate of the U.S. Army Research Laboratory at the request of the Program Manager Unmanned Ground Vehicles/Systems. Military occupational specialties examined included U.S. Army infantryman (11B), cavalry scout (19D), and the Marine Corps rifleman (0300). System required mission planning (pre mission) and navigational functions and tasks were identified. Soldier marine navigational skills were compared to mission planning and navigational tasks. Results of the analysis show that of 70 navigational skills required by the TUV system, 33 are mismatched because of a higher skills requirement, untrained system specific skills, or a combination of both.

DESCRIPTORS: *SKILLS, ROBOTICS, ARMY RESEARCH, ARMY PERSONNEL, PERFORMANCE(HUMAN), MILITARY VEHICLES, NAVIGATION, MARINE CORPS PERSONNEL, MILITARY PLANNING, MILITARY OCCUPATIONAL SPECIALTIES, TUV(TACTICAL UNMANNED VEHICLE).

AD-A339426

MICRO ANALYSIS AND DESIGN BOULDER CO

Dialogue-Based Language Training

01 Mar 1998 50 PAGES

PERSONAL AUTHORS: Plott, Beth; Suthers, Dan; Woolf, Beverly; Weinberg, Amy; Dorr, Bonnie

ABSTRACT: The U.S. Army has recognized the need to transition the research on authorable tutoring systems and language learning, both funded by themselves and others, out of the research laboratory and into more applied settings. The objective of this project was to initiate a program of research to design a computerized tutor that is capable of teaching military personnel mission relevant information and task performance through naturalistic dialog between student and tutor. The overall objective of this project was to demonstrate the feasibility of developing a general purpose, authorable, dialog based tutor. The central tasks were: (1) the design of an artificially intelligent authorable dialog system, (2) design of a lexical semantic authoring system, and (3) prototype development of an English Natural Language Processing (NLP) system. Integration issues and data exchange methods for passing information between each of these central components were also designed.

DESCRIPTORS: *ARMY TRAINING, *ARTIFICIAL INTELLIGENCE, *COMPUTER AIDED INSTRUCTION, INFORMATION EXCHANGE, COMBAT READINESS, PERFORMANCE(HUMAN), SEMANTICS, PROTOTYPES, MAN COMPUTER INTERFACE, NATURAL LANGUAGE, INDIVIDUALIZED TRAINING, CONDITIONING(LEARNING), SBIR(SMALL BUSINESS INNOVATION RESEARCH).

AD-A334196

STOTTLER HENKE ASSOCIATES INC
BELMONT CA

A Case-Based Reasoning Approach to Operator Assessment and Operator Machine Interface Enhancement

02 Jan 1998 39 PAGES

PERSONAL AUTHORS: Stottler, Richard; Davis, Alexander

ABSTRACT: In Phase 1 we investigated a Case-Based Reasoning (CBR) approach to Operator Assessment and Operator Machine Interface Enhancement for the LAMPS SH-60R Multi Mission Helicopter Upgrade (MMHU). We Developed a limited prototype case-based Operator Assessment and Operator Machine Interface Enhancement System (OA/OMIES), for the SH-60R sensor operator for a small subset of ASW situations. We developed a generic OA/OMIES architecture applicable in many other domains. The OA/OMIES tests operator knowledge through the use of tactical scenarios, derives the operators mental model based on his deficiencies revealed in the mental model. The prototype implementation provided and absolute proof by example of the feasibility of our ideas. The case-based approach offers the further benefits of automatically of semi-automatically generating the operator's mental model and of the largely circumventing the difficult and time consuming process of constructing and explicit expert mental model. Our approach could be easily extended to constitute and Intelligent Tutoring System (ITS) for the SH-60R as well.

DESCRIPTORS: *LEARNING MACHINES, *ARTIFICIAL INTELLIGENCE, *MAN MACHINE SYSTEMS, *CASE BASED REASONING, SCENARIOS, OPTIMIZATION, HELICOPTERS, MAN COMPUTER INTERFACE, OPERATORS(PERSONNEL), MENTAL ABILITY, ANTISUBMARINE WARFARE , SBIR(SMALL BUSINESS INNOVATION RESEARCH), SH-60R AIRCARFT, LAMPS(LIGHT AIRBORNE MULTIPURPOSE SYSTEM).

THIS PAGE INTENTIONALLY LEFT BLANK

Document Order Form

MAIL:
DTIC-BR
Defense Technical Information Center
8725 John J. Kingman Rd., Suite 0944
Ft. Belvoir, VA 22060-6218

CALL-IN:
DTIC-BR
Defense Technical Information Center
(703) 767-8274/(DSN) 427-8274
1-800-225-DTIC (3842)
(Menu selection 1, Option 1)

FAX:
DTIC-BR
Defense Technical Information Center
(703) 767-9070/(DSN) 427-9070

INTERNET:
msorders@dtic.mil
or
www.dtic.mil/dtic/docorderform.html

Requesting Organization Information

User Code __|__|__|__|__|__

Organization ______________________

Point of Contact ______________________

Phone Number ______________________

Method of Payment

Note: All Credit Card Types Must Be Preregistered

[ ] Deposit Account Number __|__|__|__|__-__|__

[ ] VISA [ ] MasterCard [ ] American Express

Account Number ______________________

Cardholder's Name ______________________

Expiration Date ______________________

DTIC does not accept cash, checks, or COD.

Service Codes

R – Regular Service
Priority Services (Must be called in or faxed)
P – Picked Up Next Business Day ($10.00 surcharge/document)
M – Mailed Next Business Day ($10.00 surcharge/document)
E – Express Mailed Next Business Day (Guaranteed Delivery in 2 Business Days) ($20.00 surcharge/document)

Document Orders

Service Code	AD Number	Quantity Hard Copy	Quantity Microfiche
	/JAA		
	/JAA		
	/JAA		
	/JAA		
	/JAA		
	/JAA		
	/JAA		
	/JAA		
	/JAA		
	/JAA		
	/JAA		
	/JAA		
	/JAA		
	/JAA		
	/JAA		

THIS PAGE INTENTIONALLY LEFT BLANK

DEFENSE INFORMATION SYSTEMS AGENCY
DEFENSE TECHNICAL INFORMATION CENTER
8725 JOHN J. KINGMAN ROAD, STE 0944
FORT BELVOIR, VA 22060-6218

http://www.dtic.mil
1-800-225-3842